Conceptual Modelling

A Psychological Perspective

Simon McGinnes

Submitted to meet the requirements
of the degree of Doctor of Philosophy
of the University of London

Abstract

This thesis describes the formulation and experimental use of psychological principles that apply to conceptual modelling as practised during information systems development. The principles address cognition (perception, memory and mental models) and group dynamics. The aim is to determine whether application of fundamental psychological principles can help to make modellers, especially those who are relatively inexperienced, more effective. An experimental graphical modelling technique (method 'X') is presented that conforms to the psychological principles, together with a supporting software tool for visual construction of models in the design of typical business database systems.

The effectiveness of both inexperienced and expert modellers using method 'X' in real business situations was compared with that of modellers using conventional object modelling. Data was gathered in a series of field experiments using participant observation, questionnaires, and interviews and by analysing the resulting models. With conventional object modelling, untrained modellers produced results that were grossly incomplete and incorrect (22-35%, on average). Using method 'X', untrained modellers produced models that were almost complete and correct (better than 82%). Significant productivity gains were observed with method 'X' (approximately 150% for expert modeller and over 450% for untrained modellers). For an expert modeller no measurable differences in quality were observed between methods, but the modeller regarded the quality of method 'X' models as better and expressed a preference method 'X' over the conventional approach.

The results appear to support the idea of re-engineering conceptual modelling practice according to psychological first principles. The fact that more dramatic performance improvements were observed for inexperienced modellers suggests that modelling need not require a high degree of expertise, if methods and tools are adapted appropriately. The results could be exploited to empower untrained modellers, such as end users, who wish to develop large software systems but lack access to the skills of trained IT professionals.

Contents

1 INTRODUCTION **11**

1.1 SCOPE AND AIMS OF PROJECT 11
1.2 MOTIVATION 14
1.3 BACKGROUND 17
1.4 GUIDE TO THESIS 25

2 RESEARCH DESIGN **26**

2.1 INTRODUCTION 26
2.2 RESEARCH METHODS 27
2.3 INFORMATION SYSTEMS RESEARCH 40
2.4 SELECTING APPROPRIATE METHODS 46
2.5 SELECTED METHODS IN DETAIL 54
2.6 SUMMARY 69

3 THEORETICAL FRAMEWORK **71**

3.1 INTRODUCTION 71
3.2 REPRESENTATION OF CONCEPTUAL MODELS 72
3.3 CONTENT OF CONCEPTUAL MODELS 82
3.4 CONCEPTUAL MODELLING PROCESS 94
3.5 AN ALTERNATIVE APPROACH TO CONCEPTUAL MODELLING 102
3.6 CONCLUSION 126

4 EXPERIMENTAL DESIGN **128**

4.1 STRUCTURE OF EXPERIMENT 128
4.2 MEASURES 139
4.4 SUMMARY 144

5 FINDINGS **146**

5.1 ANALYSIS OF MODELS 146
5.2 SECONDARY STUDY 209
5.3 QUESTIONNAIRE RESPONSES 209
5.4 OBSERVATIONS BY EXPERIMENTER 213
5.5 SUMMARY 217

6 INTERPRETATION OF RESULTS **218**

6.1 ANALYSIS OF EXPERIMENTAL RESULTS 218
6.2 INTERPRETATION 229
6.3 CRITIQUE OF EXPERIMENT 249
6.4 SUMMARY 255

7 CONCLUSIONS **257**

7.1 IMPLICATIONS 257
7.2 FUTURE DIRECTIONS 262
7.3 CONCLUSION 269

APPENDIX A I.S. METHODS AND CONCEPTUAL MODELLING TECHNIQUES **271**

A.1 INTRODUCTION 271
A.2 COMBINED METHODS 275
A.3 CRITICAL FACTORS APPROACH 278
A.4 DATA-ORIENTED APPROACH 280
A.5 DECISION ANALYSIS 282
A.6 NORMATIVE ANALYSIS 285
A.7 OBJECT-ORIENTED METHODS 286
A.8 PROBLEM-ORIENTED APPROACH 288
A.9 PROCESS-ORIENTED METHODS 291
A.10 PROTOTYPING 294

A.11 Socio-technical approach/participative design 295
A.12 Strategic information planning 298
A.13 Systems approach 300
A.14 Conclusions 305

APPENDIX B RELEVANT PSYCHOLOGICAL THEORY 308

B.1 Introduction 308
B.2 Memory 310
B.3 Perception and comprehension 321
B.4 Mental models 329
B.5 Group psychology 341
B.6 Conclusions 346

APPENDIX C INTERVIEW NOTES 347

APPENDIX D FORMS USED DURING EXPERIMENT 352

APPENDIX E MODELS 356

E.1 Model 1 (College Administration) 357
E.2 Model 2 (Consulting) 359
E.3 Model 3 (Distribution Warehousing) 361
E.4 Model 4 (Fraud Forum) 363
E.5 Model 5 (Fund Management) 365
E.6 Model 6 (Help Desk) 369
E.7 Model 7 (Homeopathic Remedies) 371
E.8 Model 8 (Human Resources) 373
E.9 Model 9 (International Roaming) 375
E.10 Model 10 (Legal) 378
E.11 Model 11 (Legal and Regulatory) 381
E.12 Model 12 (Mobile Phone Billing) 383
E.13 Model 13 (Operators and Networks) 385
E.14 Model 14 (Purchase Orders) 387
E.15 Model 15 (Secretariat) 389
E.16 Model 16 (Security Group) 391
E.17 Model 17 (Security/Fraud) 396
E.18 Model 18 (Stock Control) 398
E.19 Model 19 (Theatrical Productions) 400

BIBLIOGRAPHY 403

List of Figures and Tables

Figure 1.1 *The process of conceptual modelling* 11
Table 1.1 *Conceptual modelling as a candidate for process reengineering* 16
Table 1.2 *Possible interpretations of a conceptual model* 21
Table 1.3 *Chapters and appendices* 25
Table 2.1 *Summary of research dimensions* 39
Figure 2.1 *Chosen research design* 53
Figure 2.2 *Structure of experiment (after Bryman 1989)* 62
Table 2.2 *Four sets of modellers used in study* 68
Table 2.3 *Criteria for satisfactory model quality* 69
Table 2.4 *Summary of research method* 70
Figure 3.1 *Some psychological processes used in interpretation of conceptual models* 71
Figure 3.2 *Interaction of cognitive processes during perception* 72
Figure 3.3 *Mental models and conceptual models are not identical* 75
Figure 3.4 *Relationship of conceptual and mental models to business activities* 76
Figure 3.5 *Information presented in both numerical and graphical form (a)* 77
Figure 3.5 *Information presented in both numerical and graphical form (b)* 78
Figure 3.6 *Alternative ways of depicting the concept person* 80
Figure 3.7 *Conventional ways of representing aspects of the concept purchase* 82
Table 3.1 *Meaning of model constructs* 83
Figure 3.8 *Two alternative representations for the concept viewer* 84
Table 3.2 *Comparison of modelling constructs with technological concepts* 84
Figure 3.9 *Introducing a new entity type to resolve a many-to-many relationship* 85
Figure 3.10 *Two alternative versions* 86
Table 3.3 *Choice of model scope* 88
Table 3.4 *Strategies for reducing complexity* 90
Figure 3.11 *Individual and group modelling efforts* 97
Table 3.5 *Role of groups in modelling according to group size* 97
Table 3.6 *Method 'X' component types* 104
Table 3.7 *Taxonomy of method 'X' modelling constructs* 104
Figure 3.12 *Method 'X' symbols for concepts* 108
Figure 3.13 *Plural and optional concepts* 108
Figure 3.14 *Method 'X' symbols for data item types and annotation types* 108
Figure 3.15 *Searching for an image* 109
Figure 3.16 *Method 'X' model appearance* 110
Figure 3.17 *Initial notes* 111
Figure 3.18 *Beginning to identify model concepts* 112
Figure 3.19 *Structuring the model* 112
Table 3.8 *Formal meaning of model* 113
Table 3.9 *Possible approaches to requirements analysis and design* 113
Table 3.10 *Features of modelling tool* 118
Table 3.11 *Comparison with selected conventional modelling techniques* 119
Table 3.12 *Alternative views of software development* 125
Table 3.13 *Summary of principles* 127
Table 4.1 *Models* 129
Table 4.2 *Average model size by modeller's experience level and method* 130
Table 4.3 *Modellers* 132
Table 4.4 *Distribution of models by experience level of modeller* 133
Table 4.5 *Organisations modelled during study* 134
Table 4.6 *Examples of sentence correction* 136
Table 4.12 *Notational differences reconciled to allow analysis* 138
Table 4.7 *Measurements taken from models* 140
Table 4.8 *Measures used to evaluate model development* 141
Table 4.9 *Effectiveness levels* 142
Table 4.10 *Summary of measures* 145
Table 5.1 *Results for model 1* 147
Figure 5.1 *Results for model 1 (a)* 148
Figure 5.2 *Results for model 1 (b)* 149
Table 5.2 *Results for model 2* 150

Figure 5.3 *Results for model 2 (a)* *151*
Figure 5.4 *Results for model 2 (b)* *152*
Table 5.3 *Results for model 3* *154*
Figure 5.5 *Results for model 3 (a)* *155*
Figure 5.6 *Results for model 3 (b)* *156*
Table 5.4 *Results for model 4* *157*
Figure 5.7 *Results for model 4 (a)* *157*
Figure 5.8 *Results for model 4 (b)* *158*
Figure 5.9 *Results for model 4 (c)* *159*
Table 5.5 *Results for model 5* *160*
Figure 5.10 *Results for model 5 (a)* *161*
Figure 5.11 *Results for model 5 (b)* *162*
Table 5.6 *Results for model 6* *163*
Figure 5.12 *Results for model 6 (a)* *164*
Figure 5.13 *Results for model 6 (b)* *165*
Table 5.7 *Results for model 7* *166*
Figure 5.14 *Results for model 7 (a)* *167*
Figure 5.15 *Results for model 7 (b)* *168*
Table 5.8 *Results for model 8 (a)* *169*
Table 5.8 *Results for model 8 (b)* *170*
Figure 5.16 *Results for model 8 (a)* *170*
Figure 5.17 *Results for model 8 (b)* *171*
Figure 5.18 *Results for model 8 (c)* *172*
Table 5.9 *Results for model 9* *173*
Figure 5.19 *Results for model 9 (a)* *174*
Figure 5.20 *Results for model 9 (b)* *175*
Table 5.10 *Results for model 10* *176*
Figure 5.21 *Results for model 10 (a)* *177*
Figure 5.22 *Results for model 10 (b)* *178*
Table 5.11 *Results for model 11* *179*
Figure 5.23 *Results for model 11 (a)* *180*
Figure 5.24 *Results for model 11 (b)* *181*
Table 5.12 *Results for model 12* *182*
Figure 5.25 *Results for model 12 (a)* *183*
Figure 5.26 *Results for model 12 (b)* *184*
Table 5.13 *Results for model 13* *185*
Figure 5.28 *Results for model 13 (b)* *187*
Table 5.14 *Results for model 14* *188*
Figure 5.29 *Results for model 14 (a)* *189*
Figure 5.30 *Results for model 14 (b)* *190*
Table 5.15 *Results for model 15* *191*
Figure 5.31 *Results for model 15 (a)* *192*
Figure 5.32 *Results for model 15 (b)* *193*
Table 5.16 *Results for model 16* *194*
Figure 5.33 *Results for model 16 (a)* *195*
Figure 5.34 *Results for model 16 (b)* *196*
Table 5.17 *Results for model 17* *197*
Figure 5.35 *Results for model 17 (a)* *198*
Figure 5.36 *Results for model 17 (b)* *199*
Table 5.18 *Results for model 18 (a)* *200*
Table 5.18 *Results for model 18 (b)* *201*
Figure 5.37 *Results for model 18 (a)* *201*
Figure 5.38 *Results for model 18 (b)* *202*
Figure 5.39 *Results for model 18 (c)* *203*
Table 5.19 *Results for model 19 (a)* *203*
Table 5.19 *Results for model 19 (b)* *204*
Figure 5.40 *Results for model 19 (a)* *204*
Figure 5.41 *Results for model 19 (b)* *205*
Figure 5.42 *Results for model 19 (c)* *206*
Table 5.20 *Summary of results for main study (experienced modeller)* *206*

Table 5.21 *Summary of results for main study (inexperienced modeller)* *207*
Table 5.22 *Average measures for different modellers using each technique* *207*
Table 5.23 *Differences between results for method 'Y' and method 'X' (Δ = X - Y)* *208*
Table 5.24 *Change frequencies* *208*
Figure 5.43 *Change frequencies* *209*
Table 5.25 *Results of secondary study* *210*
Table 5.26 *Summary of questionnaire responses by modelling session participants* *210*
Table 5.27 *Summary of questionnaire measures* *211*
Table 5.28 *Average questionnaire ratings by perceived purpose of modelling* *211*
Table 6.1 *Modeller groups* *220*
Figure 6.1 *Completeness vs. modeller's experience level* *221*
Figure 6.2 *Correctness vs. modeller's experience level* *222*
Figure 6.3 *Productivity vs modeller experience level* *223*
Figure 6.4 *Error rate vs modeller experience level* *223*
Table 6.2 *Error frequencies* *224*
Figure 6.5 *Error frequencies* *225*
Table 6.3 *Error frequencies for expert modeller* *225*
Table 6.4 *Summary of effectiveness levels achieved* *226*
Table 6.5 *Correlation of effectiveness facets for all modellers* *226*
Table 6.6 *Correlation of effectiveness facets for inexpert modellers* *227*
Figure 6.6 *Effectiveness vs modeller experience level* *227*
Table 6.7 *Rounded average effectiveness scores (as % of theoretical max. effectiveness)* *228*
Figure 6.7 *Typical pattern for an experienced modeller using both methods* *231*
Figure 6.8 *Typical pattern for an experienced modeller using both methods* *232*
Figure 6.9 *Typical pattern for an inexperienced modeller using method 'X'* *233*
Figure 6.10 *Typical pattern for an inexperienced modeller using method 'X'* *233*
Figure 6.11 *Typical pattern for an inexperienced modeller using method 'Y'* *234*
Figure 6.12 *Typical pattern for an inexperienced modeller using method 'Y'* *235*
Table 6.9 *Patterns observed for experienced and inexperienced modellers using both methods* *236*
Table 6.10 *Summary of typical performance for each type of modeller* *236*
Figure 6.13 *No. of relationships vs. no. of components* *237*
Figure 6.14 *No. of relationships vs. no. of changes (excluding model 5)* *238*
Table 6.8 *Summary of potential sources of bias and experimental precautions* *254*
Table 7.1 *Possible alternative approach to system development* *260*
Table 7.2 *Roles* *265*
Figure 7.1 *Modelling-generating-use cycle* *268*
Table A.1 *Classification of methods* *272*
Table A.2 *Reference information systems life cycle* *273*
Figure A.1 *Relationships between method components and evaluation categories* *274*
Figure A.2 *SSADM entity life history diagram (part) (after Ashworth and Slater (1993) p.91)* *277*
Table A.3 *SSADM event/entity matrix* *277*
Figure A.3 *Meta-model for CSF analysis* *279*
Figure A.4 *One-Page Management focus report* *280*
Figure A.5 *Information Engineering entity diagram* *282*
Table A.4 *Unfounded assumptions about information cited by Ackoff* *283*
Figure A.6 *Meta-model for Ackoff's approach* *284*
Figure A.7 *BICS organisation versus data class matrix (after Zachman (1982) p.41)* *286*
Figure A.8 *Object-Oriented Analysis Class-&-Object diagram* *288*
Figure A.9 *ISAC A-graph (after Lundeberg et al (1982) p.18)* *290*
Figure A.10 *ISAC I-graph (after Lundeberg et al (1982) p.27)* *291*
Figure A.11 *Structured analysis data flow diagram* *292*
Figure A.12 *Structured analysis state-transition diagram* *293*
Table A.5 *Battaglia's strategic information planning steps* *298*
Table A.6 *BSP process/data matrix* *300*
Figure A.13 *SSM conceptual model (after Checkland and Scholes (1990) p.71)* *302*
Figure A.14 *SSM rich picture (after Checkland and Scholes (1990) p.47)* *303*
Figure A.15 *Maltese Cross (after Wilson (1990) p.239)* *305*
Table B.1 *Relevant psychological areas* *308*
Figure B.1 *Relevant psychological topics* *309*
Figure B.2 *Simple model of memory* *311*

Table B.2 *Schank's memory types* *314*
Figure B.3 *Simple categorisation hierarchy.* *316*
Figure B.4 *Hierarchical model of semantic memory* *317*
Figure B.5 *Spreading activation model (Best 1999)* *318*
Figure B.6 *Chunking* *320*
Figure B.7 *Gestalt laws of perception* *323*
Table B.3 *Types of set* *329*
Figure B.8 *Taxonomy of internal knowledge representations (Eysenck and Keane 1995)* *331*
Table B.4 *Schank's conceptual primitives* *332*
Figure B.9 *Simple conceptual graph (Sowa 1984)* *334*
Figure B.10 *Conceptual graph for concept BUY* *335*
Figure B.11 *Concept BUY in use* *335*
Table B.5 *Text before restructuring* *339*
Table B.6 *Text after restructuring* *340*
Table B.7 *Types of task* *342*

Acknowledgements

I would like to thank Dr James Backhouse for his invaluable help and support as my supervisor throughout this project. I would also like to acknowledge the assistance I have received from many others who contributed personally to the ideas in this thesis during many interesting and productive conversations: Johnny Amos, Gina Boreham, Sarah Doherty, Patrick Fillatre, Ollie Guinan, Aisling Guy, Terry Halpin, Dierdre Hurley, Marian Kelly, Jayne Kent, Mireille Kocko, Fabien Lessirard, Joanne McDowell, Elizabeth McGinnes, Liz McGinnes, PJ McKenna, Lisa McMahon, Gerry Mooney, Rosemarie O'Mahony, Ray O'Neill, Mark O'Neill, Muireann Ridge, Gloria Santamaria, Adrian Stokes, Erwan Thepaut, Mark Treanor, Vincent Visdeloup and Dave Warren. Special thanks are due to Joanne McDowell for her dedication in making the modelling tools work so effectively. And, of course, thanks to my wife Fiona for her help and patience.

"I will contend that conceptual integrity is the most important consideration in system design."

Fred P. Brooks

"A complex system that works is invariably found to have evolved from a simple system that works."

John Gaule

"The ability to simplify means to eliminate the unnecessary so that the necessary may speak."

Hans Hofmann

"As a rule software systems do not work well until they have been used, and have failed repeatedly, in real applications."

Dave Parnas

"It is hard to let old beliefs go. They are familiar. We are comfortable with them and have spent years building systems and developing habits that depend on them ... ... we forget that the world looks to us the way it does because we have become used to seeing it that way through a particular set of lenses. Today, however, we need new lenses. And we need to throw the old ones away."

Kenich Ohmae

1
Introduction

1.1 Scope and aims of project

This project is about conceptual modelling, as practised during the development of information systems. It involves a critical examination of existing conceptual modelling methods, formulation of useful principles that apply psychological thinking to conceptual modelling, and experimentation in practical situations with a conceptual modelling method that uses the principles. The aim is to test the idea that application of psychological first principles can help to improve the effectiveness of conceptual modellers. The research focuses on how conceptual modelling is done and who does it. Other factors, such as its context, why it is done, what it uses as input, and what it produces, are also considered. To make the scope of the research clear, Figure 1.1 depicts conceptual modelling in context. In very broad terms the subject matter for this research is bounded by the edges of the process.

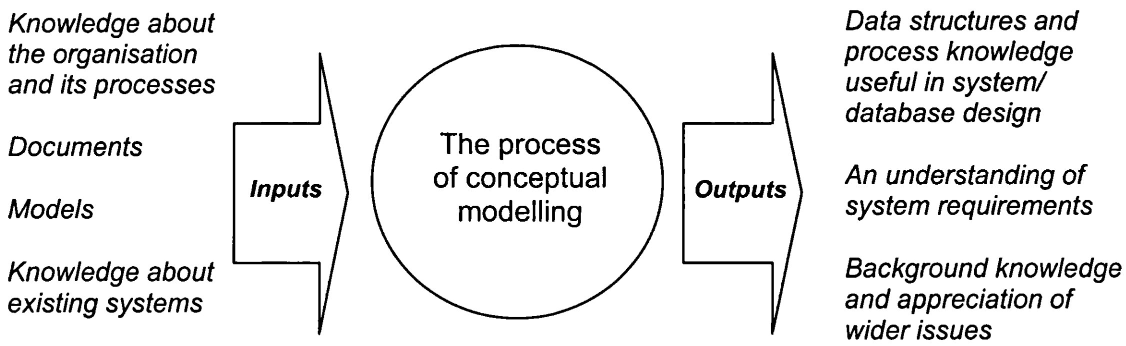


Figure 1.1 The process of conceptual modelling

The process of conceptual modelling is inextricably linked with the processes of information system design, construction and use. Therefore we shall also consider these processes to some extent. Some would also claim that it has a role in strategic business planning and even in business management. However, in order to retain a tight focus we shall not devote much attention to these applications of conceptual modelling. Appendix A contains details of a number of information systems development methods and associated conceptual modelling techniques. It is clear that conceptual modelling techniques come in many forms. To avoid doubt, we shall use in this thesis two working definitions of conceptual modelling, and we shall use the terms 'conceptual model' and 'conceptual modelling' (see below) with both senses. Where context does not make it obvious, the intended meaning will be stated explicitly. Throughout the text, in the absence of qualification, the wider definition may be assumed.

Narrow definition In the narrow sense, the term 'conceptual model' refers to any object or data model produced during the development of software, when used primarily to represent business concepts (as opposed to software concepts). Although some definitions of conceptual modelling exclude object models, they are included here because of their similarity to data modelling methods when viewed in the wider context (when compared with the broad range of alternative modelling techniques). Conceptual modelling techniques conforming to our narrow definition are techniques geared specifically (but not necessarily exclusively) to capturing definitions of concepts that are meaningful to the business users of information systems. They capture concept definitions chiefly because information systems must accept and store data relating to the business. The concept definitions are typically expressed in an abstract way (e.g. in the form of object classes or entity types). This narrow definition of conceptual modelling includes popular techniques such as UML class diagrams (Fowler and Scott 1997) and entity relationship modelling (Chen 1976), provided that they are used at a business level, to model business concepts rather than software structures exclusively.

Wide definition In the wider sense, a conceptual model is any representation of business information which can contribute to an understanding of business concepts that may be useful in the design of information systems. A range of systems analysis and design techniques are used to capture business-related information during the design of information systems, and many of these capture information about business concepts. For example, dataflow diagrams can imply the existence of certain business concepts. They qualify as conceptual models under the wide definition because their data flows and data stores represent collections of data items that would be useful in the definition of business concepts. But dataflow diagrams do not qualify as conceptual models under the narrow definition because they incorporate no means of defining the business concepts directly.

Both the narrow and the wide definitions refer to *business concepts*. We define the term 'business concept' to refer to *the idea of* something that an information system has to store data about or be otherwise aware of: an information system may have to store data relating to certain types of people, organisations, documents, activities, and so on. This definition excludes things that information systems do not need to store information about or be otherwise aware of, and the distinction is the basis for our idea of *relevance*. We discuss the idea of relevance further in later chapters (for example, see Section 3.5).

In this thesis when we talk about storing data (or information) about a business concept, we mean storing one or more items of data for each instance of the concept. The data may have more or less internal structure. For example, for the business concept 'employee', we may allow for storage for each employee of a name (a structured text string), a salary figure (a money amount), a department (a link to a particular organisational unit, which has its own data), a curriculum vitae (a word processor document) and so on. This means that the data we store about concepts can take many forms, including simple data items (text, numbers, dates, etc.), structured data items (names, addresses, etc.), links to other concepts, and "unstructured" data (such as word processor documents).

A *business concept definition* is defined as a statement of the data that an information system must store in relation to a particular business concept. Concepts are inevitably interrelated and so any business concept definition must include links to related concepts. Note that our definition of 'business concept definition' means that it is not sufficient to define business concepts informally (e.g. using prose, as in a dictionary). Informal concept descriptions such as prose descriptions may well be expressed in terms of other concepts, but they incorporate no formal links to those concepts. We accept this form of concept definition only as an intermediate stage leading to a 'full' definition, in terms of other concepts, as described above. However, we note that any business concept definition may incorporate various items of descriptive material that help to explain the concept, as opposed to defining it formally. For example, my definition of 'employee' might include a picture of a 'typical' employee and a sample job specification.

Any modelling technique can be classified using these working definitions. For example, according to our definitions, program flowcharts may not be considered as conceptual models because they do not allow direct definition of business concepts (narrow definition) and they do not capture any significant information that would be useful in defining business concepts (wide definition). Rich pictures and the 'conceptual models' of Soft Systems Methodology (Checkland 1981) can be considered conceptual models because they meet the requirements of the wide definition when used in connection with software design. However, they do not conform to the narrow definition and, in fact, the sense in which Soft Systems Methodology uses the term 'conceptual model' is quite different from our narrow definition.

Finally, we define the *process of conceptual modelling* as any activity that involves the construction of conceptual models, whether they meet the narrow definition or the wider definition. This definition admits a wide variety of situations including group modelling sessions using Joint

Applications Design (Wood and Silver 1995) as well as lone modelling by individuals who wish to design databases.

1.2 Motivation

This work springs from a perception that the practice of conceptual modelling (in the wider sense) could benefit from some new thinking. In this context, conceptual modelling means the work—often done by systems analysts or business analysts—that aims to capture information about an organisation, its data and its processes, in a form suitable for use during the development of computer-based information systems. The perception is that present-day conceptual modelling practice tends to exclude business end users, who are its main customers. It requires a high level of expertise and may be practised only by skilled professionals. Understanding conceptual models can be difficult. Yet, well-designed custom business applications cannot be produced without some form of conceptual modelling. Conceptual modelling, as currently practised, can be seen as a barrier that prevents less experienced people from specifying and creating the business applications they need or want.

Conceptual modelling has two main purposes. It allows the information needed for good system design to be structured and ordered (Marakas and Elam 1998). But it also provides a medium for communicating this information between system users and system developers, allowing review, discussion and refinement. It is this second purpose—communication—that we shall concentrate on. One starting point for this thesis is the observation that models constructed using conventional conceptual modelling techniques may not be an ideal way of communicating.

Conceptual models may be expressed in a variety of forms, but most typically are depicted using diagrams. Object models, entity-relationship diagrams, use case diagrams, role-activity diagrams, data flow diagrams, state-transition diagrams and entity life histories can all be considered to be types of conceptual model in the wider sense given above. These tried and tested diagrammatic modelling techniques have developed over many years, to the point where they form an established part of information systems practice. Innovations such as object-orientation, relational database management systems and workflow systems have often gone hand-in-hand with associated diagram-based modelling techniques.

A common assumption is that conceptual modelling, and hence 'real' system development, can be practised only by expert and highly trained analysts or designers. According to this view, end users may be able to construct small, simple systems, but any reasonably complex system must be designed by a professional. CASE tools evolved—as support for experts—in response to this perception (King 1996). But, increasingly, the distinctions between the providers and the consumers of software are blurring. End users may now quickly construct complex database systems, spreadsheet applications, and world-wide web sites, using cheap and user-friendly development tools. As recently as the early 1990s it was generally held that hands-on computing required a good deal of expertise and training. Considering the nature of information technology at the time this view may have been justified. But life has changed. Many end users now embark on system development without software development expertise and without reliance on the skills of IT specialists. The term *end user* is itself increasingly misleading and inappropriate (but is used in this thesis in the absence of any widely-understood alternative).

Conceptual modelling techniques have not reflected the trend towards increasing ease of use. They still require expertise. With the rise of end-user computing it can be argued that conceptual modelling itself has become irrelevant, at least for smaller systems. End users don't need to bother with conceptual models for small applications. They can rely on their own judgement and on software tools with wizards and similar aids. The short-term cost of poor design has lessened, for smaller systems development projects. But the long-term cost is probably just as high, when maintenance, redesign and the inconvenience of using badly designed systems are taken into account. And conceptual modelling is still a necessity for medium-sized or large developments, that remain the responsibility of IT professionals. It is the small to medium-sized projects, conducted by less experienced staff, that we hope will gain most from the results of this research. We hope to gain insight into ways of reducing the level of expertise needed to create models, and hence to reduce the need for dependence on skilled IT staff, perhaps even empowering end users themselves to construct models and to use them to design and deploy non-trivial, but well-designed, systems.

A case for change

Present-day conceptual modelling techniques arguably do not lend themselves to current working methods such as RAD (Stapleton 1997) and JAD (Wood and Silver 1995). They are 'technical', which can be off-putting for end users (Bansler and Bødker 1993). It is difficult to use them with the speed and flexibility demanded by rapid application development methods. The Unified Modelling Language (UML), one of the latest modelling techniques (Fowler and Scott 1997), illustrates this point. UML includes many different notational devices and a solid grounding in the concepts and principles of object-orientation is necessary to use it. As with most conceptual modelling techniques, its diagrams can be complex and restructuring them is often onerous. Today's conceptual modelling practice has evolved over a period of years, with contributions from diverse sources. The process of modelling itself was never explicitly designed by any individual or group and, like many ad hoc business processes, may apply outmoded rules and assumptions. Under changed circumstances, methods that were once useful can start to threaten effectiveness. Table 1.1 lists some common symptoms of ineffective business processes (Hammer and Champy 1993) and identifies their relevance to conceptual modelling practice, highlighting a clear prima facie case for improvement. Process improvement often involves the introduction of information technology in one form or another. In business process reengineering (BPR) information technology is seen a disruptive force that has the *"ability to break the rules that limit how we conduct our work"* (Hammer and Champy 1993). Its use in process improvement hinges on this destabilising effect (Ciborra and Jelassi 1994).

Symptom	**Relevance to conceptual modelling**
Complexity	Conceptual modelling requires extensive training and experience to be practised well. Models use complex notations and different diagram types may be interrelated in complex ways.
Extensive information exchange	Information must be communicated by end-users to an analyst, who translates the information into logical model form, and then translates the model back into the users' terms so that they can check its accuracy. The logical model must be converted into system specifications (including a physical model, if used) that technical specialists can work with.
Data redundancy	The same information may be encoded many times in different models (for example, the concept *purchase* may be represented in data model as a data entity but in a process model as both a process and an information flow). Facts may need to be expressed in prose (e.g. for end users), in logical model form, and in technical form for technical specialists.
Rekeying	Models created on whiteboard or flipchart must be transcribed onto paper and perhaps keyed into a CASE tool. Later, the contents of the model may be re-keyed into development tools.
A high ratio of checking and control to value-adding	Models of different types must be checked against one another (e.g. data models checked against process models). The analyst must ensure that users understand models sufficiently well to be able to check them against their own view of the business. Effort must be devoted to preserving formal correctness in models while the benefits of doing so are not always clear.
Poor quality end results	The quality of the end result is highly dependent on the skill and insight of individual analysts. Less experienced analysts can produce very poor quality models.

Table 1.1 Conceptual modelling as a candidate for process reengineering

It is a measure of the entrenched nature of current practice that many experienced IT professionals may be unaware that their processes are candidates for improvement, despite the attention paid to software process improvement in recent years. In a sense, the conceptual modelling die was cast in the 1970s when structured methods came into vogue. It can be argued that today's methods are very similar to those early structured techniques. The techniques of the 1970s evolved before automated support for the modelling process was feasible and before today's development tools existed. Sophisticated CASE tools are now available to support the conceptual modeller, but present-day tools fail to match the reality of the modelling task (Connell and Shafer 1995). If most of the thinking about models is done away from the tools then they cannot be said to provide meaningful support to the modelling process (King 1996). Process improvement in conceptual modelling presumably, therefore, could make better use of the rule-breaking capabilities of information technology.

Why is psychology relevant to conceptual modelling?

As we shall see in Chapter 3, psychological ideas support in many ways the idea of rethinking the conceptual modelling process. From a psychological perspective, conceptual modelling techniques appear to neglect much of what is known about the cognitive processes of perception and comprehension. They do not utilise our innate and automatic visual recognition abilities and they place unrealistic demands on memory (Solso 1998). Modelling techniques seek to capture end users' knowledge about business systems, but fail to offer any direct correspondence to their mental models (Johnson-Laird 1993). Consequently, using them can require great mental effort. Psychology therefore seems a good place to look for some new ideas. In a sense, it is the most fundamental relevant discipline if one is interested in creating a process that can help individuals to work together and to share and refine complex ideas. Psychology is the science of perception and comprehension and it encompasses the highly relevant fields of mental models, cognition and group behaviour. It is an obvious first port of call in a search for fundamental principles, which until now have been lacking, to guide the improvement of conceptual modelling.

One may ask why other disciplines are not mentioned. Conceptual modelling is a multi-dimensional activity and many fields are potentially relevant. Psychological, management, business, sociological and technical aspects are all of interest. The answer is that, while the chief focus in this thesis will be on psychology, relevant knowledge and perspectives from other disciplines will be used if they can help to shed useful light on the subject matter.

1.3 Background

Conceptual modelling techniques

Appendix A contains a historical review of information systems development methods and their associated conceptual modelling techniques. An often-stated aim of using conceptual modelling (in the wider sense) during the development of computer-based information systems is to help analyst and user come to a shared understanding of a business area (Loucopoulos and Zicari 1992). In theory, conceptual modelling techniques allow us to capture information about organisations in an accessible but structured form. Conceptual modelling techniques are very often based around diagrams, which are often held to be easier to understand by analysts and users. For instance, *"the chief merit in a diagrammatic technique is in user communication"* (Olle et al 1991).

A standardised, formal way of representing the information in conceptual models is considered necessary to permit verification tasks, such as cross-checking, to be carried out. It may further be assumed that the use of conceptual modelling techniques tends to produce a better end result (i.e. a better information system). Justifications cited for this view include the claim that particular modelling techniques are better because they are more structured and more formal and even, in some cases, mathematically-based (e.g. Halpin and Nijssen 1995). According to one influential work, a fundamental requirement for conceptual modelling techniques is that *"descriptions should be stated in a formalism with unambiguous syntax which can be understood by a suitable processor"* (Loucopoulos and Zicari 1992).

But little hard evidence is presented to support these claims. The view that formality is an important priority for conceptual modelling techniques has for some time been questioned (Vitalari 1984, Bansler and Bødker 1993, Pyshlin 1991). Experience indicates that the use of conceptual modelling techniques, especially when end-users are involved, is not as simple as conventional wisdom might suggest. Diagrams are often not understood. The more formal a model becomes, the less useful it seems as a vehicle for communication. Even the most vociferous advocates of structured methods (in the wider sense) acknowledge that models constructed using up-to-date diagrammatic techniques can be hard for end users to understand (Martin 1993). In the past many 'theories' in information systems have essentially been normative solutions with little or no theoretical or empirical justification (Remenyi and Williams 1996). Evidence suggests that a gulf exists between *"the way systems development is portrayed in the mainstream of scientific and technical literature and the way it is carried out in real life"* (Bansler and Bødker 1993). The same authors observe that much research in systems development rests on a rather simplistic, and often misleading, understanding of the nature of system development.

The goal of this research is to develop a view of conceptual modelling—one particular aspect of information systems development—that is built upon a firm theoretical and empirical base. To be able to do this we must explore the processes involved in conceptual modelling, whether they are internal psychological processes or external social processes. An understanding of these processes will help us to appreciate the behaviour of IT specialists and their clients as they work together to construct and use conceptual models (King 1996). Several avenues could fruitfully be explored. A study of the organisational and political processes involved in conceptual modelling might shed useful light on how and why the techniques are used. The role of the conceptual model as a vehicle for consensus-building and organisational change could be explored. We could develop the idea of conceptual modelling techniques, and of related technologies such as information systems methods and CASE, as political tools used by IT specialists to retain and wield power within organisations. These are important ideas that deserve further treatment (Beath and Orlikowski 1994). Related research work can be found in the literatures on management, organisational behaviour, human resources and anthropology.

Ideas of this sort inevitably form part of any discussion of conceptual modelling techniques and, to a limited extent, they will be introduced in this work. But this research is not an attempt to understand conceptual modelling in totality, which would be impractical. Instead we concentrate on the 'surface' goal of conceptual modelling: to allow information about an organisation to be represented in an accessible but structured form, that can be understood and verified by end users while at the same time being technically useful in the design of information systems. Taking this goal at face value, the current research is aimed at finding ways of improving existing conceptual modelling practice, so that this goal may more easily be attained. To help ensure that the research is bounded in scope, in the experimental work (Chapters 4 and 5) we shall consider mainly conceptual modelling in the narrow sense as defined in section 1.1. However, for the time being we shall continue to use the second, wider definition of conceptual modelling.

The philosophical standpoint of this research is essentially pragmatic and optimistic, and not unlike that held by researchers in human-computer interface (HCI) (Nielsen 1994). HCI research takes as given that computers are used and assumes that they can be made more useful through good design, empowering end users by enriching the tools available to them. In a similar vein, this research does not question why systems are developed using conceptual modelling techniques but, instead, asks how the modelling process can be improved and enriched, empowering and benefiting the people and organisations that participate in it. We take as given that conceptual modelling can be useful and that it is practised in good faith, with the intention predominantly being to create better quality information systems. But we do not assume that existing practice necessarily achieves the best possible results.

End-user participation

It is common for end users to be involved in the creation and verification of information systems generally and conceptual models specifically. Many information systems methods recommend it (Jacobson 1991, Martin 1990, Wood and Silver 1995). Well-established opinion holds that users should be heavily involved in the whole of the systems development process (Pasmore 1988, Stapleton 1997). However, little research has specifically assessed the strategy of involving users in conceptual modelling. Some researchers have pointed out the underlying contradictions inherent in information systems methods that advocate end user involvement (Beath and Orlikowski 1994). It seems logical for users to verify the content of conceptual models since the models are supposed to represent the users' own business concepts. However, there is evidence that IT practitioners avoid presenting conceptual models to users, except in simplified form. One study found that designers believe that users are unable to understand models such as data flow diagrams. A reason the designers gave was that conceptual models were considered to be based on an 'information processing' view rather than one that users were familiar with (Bansler and Bødker 1993). Conceptual models may be seen more as an expression of intentions regarding system design than as a statement of the structure of business concepts. In this work we assume that users are likely to be involved in most conceptual modelling efforts, alongside IT professionals, to ensure that models match business concepts, processes and requirements. Obviously this has implications for the kinds of model that can, or should, be used.

Modelling business concepts?

A fundamental question is whether or not conceptual models represent users' business concepts. It can be argued that the most widely-used contemporary modelling techniques concentrate on information that will be directly useful during design and ignore facts not translatable into designs (McGinnes 1994). For example, object modelling techniques represent object structures in a form that is very close, if not identical, to the way in which software objects are implemented. Transformation of a model into software structures or physical model normally involves only a small number of steps, such as the conversion of associations into classes or pointers and the merging of classes in inheritance hierarchies. A physical model may subsequently undergo a considerable amount of tuning, but the original object model and the subsequent physical model remain closely related. A similar situation exists with respect to conversion of data models into database schemas. The first-cut conversion process is so simple that many CASE tools perform it without intervention.

Similarity between logical and physical models is a pragmatic choice since a main aim of using modelling techniques is to help in the design of software structures—programs and databases. It is sometimes claimed that modelling techniques offer an intuitive, user-orientated view of 'real-world' objects (e.g. Barker 1990, Herbst 1997). But if conceptual models are so close in structure to software systems, is it realistic to suggest that they can also match business reality? (Hanseth and Monteiro 1994). To help explore this question several alternative views of conceptual models are listed in Table 1.2. Although the different perspectives are quite distinct from one another, we note that many methods do not clearly indicate where they fit in this scheme. View (a) has the advantage of allowing accurate capture of business facts, but suffers from the lack of any easy route into system design. Separate business and software models would presumably be required. Process modelling methods such as STRIM (Ould 1995) fall into this category, although it is arguable that they distort reality because of the need to use a fixed and therefore restricted notation. View (b), in the absence of any supporting requirements definition, could be characterised as a 'software engineering' approach rather than an information systems approach. It is suited to the design of software in isolation but allows no explicit connection to be made between the software structure and the business world. Program flowcharts and pseudocode fall into this category. View (c) is probably the most common. It can be seen as efficient, since only one set of models is required. Many modelling techniques attempt to represent business information in a way that lends itself towards implementation, even if it is not directly capable of being implemented. Traditional methods such as SSADM and structured analysis are based on this view—although they may claim to be based on view (a). Object-oriented methods like UML also fall into category (c). The object representation is often described as a natural one for business requirements (Coad and Yourdon 1991). However, a problem with this approach is that it can be difficult to formulate model concepts that represent both software and business concepts at the same time. Software structures, whether object-oriented or not, may make poor conceptual models. We shall discuss this issue briefly below, before moving on to consider view (d).

View of model	Explanation: a conceptual model is ...
a. As a business model	An objective statement of business reality.
b. As a software model	An objective statement of the structure of a software system.
c. As a business-cum-software model	An intermediate stage between (a) and (b) above (i.e. between the user's business perspective and a physical system design). The model must be understandable by both users and developers and is a compromise that encapsulates sufficient information about business concepts to be verified by a user, but structures this information in a way that is useful to the developer.
d. As a model of mental concepts	A subjective statement of one or more users' view(s) of business reality as they perceive it.

Table 1.2 Possible interpretations of a conceptual model

Software structures as conceptual models

Modelling techniques of type (c) attempt to mimic software structures while at the same time modelling the underlying business organisation, information or processes. Any software that is used as part of a business process is, in some sense, a model of the process. Reverse engineering has developed because useful facts about an organisation—'business rules', constraints, data structures, and so on—can be gleaned from the structure of existing databases and programs. But it is hard to recover this knowledge. Software is often modified because of implementation issues (e.g.

performance) and its structure may therefore be misleading if interpreted literally as a reflection of a business system. Effective reverse engineering depends heavily on both the skill of the engineer and the use of sophisticated tools (Kudrass, Lehmbach, Buchmann 1996).

Prototyping can also be viewed as a form of conceptual modelling in which the user 'reads' a prototype system to check its match with business reality (Mathiassen, Seewaldt and Stage 1995). It is probably easier to deduce an underlying business process from the user interface of a piece of software than by examining the software's internal structure. However, choice of user interface design is itself influenced heavily by external factors such as design standards or conventions, and any user interface can be only an incomplete and ambiguous representation of the business in which it is used.

Hence we conclude that a piece of software can be viewed as a model of a business system but that it will inevitably have serious limitations as a conceptual model, and that unlocking its information will not always be practicable or easy. This suggests that it may also be difficult or risky to use conceptual modelling techniques that mimic the internal structure of software systems (view c). We may have difficulty in relating the conceptual structures expressed in our models to our own business-related concepts (Hanseth and Monteiro 1994). The model may be forced into a particular form for reasons that have nothing to do with business reality. Consequently it may be more advisable to adopt structures and language closer to the user's own business world.

Subjectivity

We now consider view (d) in Table 1.2, which refers to subjective mental concepts. Subjectivity is discussed here because it is a common theme running through this work. Subjectivity is fundamental to understanding conceptual modelling itself. It is also relevant to the research methods that we shall adopt (see Chapter 2). The ontological standpoint of this work is that our experience of reality is inevitably subjective and highly dependent on context. We may assume that our senses give us access to reality. But whether an external objective reality exists is in fact irrelevant, because we can never have access to it. All experience of the world is mediated by perceptual mechanisms that guarantee unconscious distortion and interpretation. It is impossible to experience the external world, or indeed our own thoughts and memories, without interpretation. Because of the way the mind works, the act of experiencing *is* interpretation (Best 1999). This issue has deep importance to conceptual modelling and is explored in more detail in Appendix B. This fact about an individual's perception guarantees that group work can never be a situation in which people agree on an external objective reality, whether it be in business organisations, politics, education or any other sphere. If people do agree, it must be on some other 'reality'. Add to this the nature of group dynamics and politics, which provide the context for most work in organisations, and it is easy to see that shared perceptions can never be taken as evidence of an objective reality. Of course, there is a shared physical world containing people, places and physical objects (or, at least, this is a good working approximation). But everything else is interpretation. Many of the things that form the day-to-day business of organisations (such as business plans, employment, equity, organisational structures and business transactions) exist only in the mind and hence rely entirely on consensus. If there were an objective real world, they could not be part of it.

Hence this research is concerned with subjectivity. The rationale above has been stated in psychological terms and may therefore seem unfamiliar to information systems researchers who might be more used to talking about *social construction of reality* (Berger and Luckman 1966). But the point is the same one. We must acknowledge that every individual has his or her own unique perception of the world and that organisational 'truths' exist by agreement only. The information systems literature contains many coherent and thoughtful explorations of this idea and its implications (e.g. Walsham 1993), although they are more often stated in the language of the social sciences. We shall not reproduce those arguments here. But we must explore the consequences of this inherent subjectivity for the present research. It affects the research methods we shall use (see Chapter 2). It also has an impact on conceptual modelling itself, which is discussed below.

What really happens when people produce conceptual models?

Conceptual modelling is often thought of as an exercise that involves the capture of facts about reality (Lewis 1993). This could be termed a realist or objectivist point of view. But common experience of conceptual modelling runs counter to this view. Conceptual models normally represent the business *as it is seen by* one or more end users. Very often, the modelling process is itself responsible for analytical development of concrete ideas where previously there was only intuitive understanding. Participants in conceptual modelling sessions often claim (and did so in this study) that the act of modelling is of itself beneficial since it causes them think systematically and analytically about the business they work in, something they may not have done before. The essence of modelling consists of teasing out the users' own views about the business. This is the subjective, nominalist position originally identified by Burrell and Morgan (1979). Only a naïve analyst would attempt to impose a supposedly objective outsider's view. In fact, Chen's influential (1976) paper on entity-relationship modelling clearly stated that this type of modelling was concerned with representing mental concepts, not real-world concepts. It is only in the more recent literature that the idea of modelling 'reality' has crept in (Rumbaugh et al, 1991).

There is evidence that *"practitioners understand the implicit non-objectivist issues (in modelling), but that researchers tend to ignore them"* (Hitchman 1997, Veryard 1992). One cannot work for long with conceptual models in organisations without realising that modelled reality should be what people say it is and not some externally judged objective reality (Hitchman 1997). In many ways a good modeller is like a conduit for the business end users' ideas and knows to steer clear of 'fixing' the model to make it more like his or her own idea of external reality. If a modeller 'fixes' the model at all, it is simply so that it reflects more closely the perceived intentions of the relevant people and, where necessary, achieves a reasonable compromise. The IT professional must accept a model as correct if the relevant business users feel that it is correct, with the proviso that he or she also has a responsibility to ensure that any resulting information systems function effectively as far as the end user is concerned.

Development of ideas leading to this research

This work follows a period of some years in which the researcher operated as both a consultant and an academic in the information systems field. The modelling tool and many of the ideas described in the first half of this thesis emerged during this period of reflective practice, in which the author was immersed in conceptual modelling practice in large and small organisations, whilst teaching information systems material across a broad spectrum and conducting research into information systems methods, psychology and related areas. This period was not in any sense a formalised or rigorous research project, but it led to the development of, and created the context for, the work described in this thesis. This work is best understood if viewed as the culmination of a long gestation period of practice and reflection that led ultimately to the development of the ideas behind the research.

1.4 Guide to thesis

This thesis is organised as follows:

Section	**Title**	**Content**
Chapter 1	Introduction	An overview of the research objectives together with background material.
Chapter 2	Research design	An outline of the research methods used.
Chapter 3	Theoretical framework	An account of how psychological theories and research results can be applied to conceptual modelling. A series of principles is derived and an experimental conceptual modelling method is presented that conforms to the principles.
Chapter 4	Experimental design	A description of the experimental portion of the research, in which the experimental modelling method was applied in practical situations.
Chapter 5	Findings	The results of the experimental work.
Chapter 6	Interpretation of results	An interpretation of the experimental results.
Chapter 7	Conclusions	Concluding remarks.
Appendix A	I.S. methods and conceptual modelling techniques	An overview of past and present conceptual modelling approaches.
Appendix B	Relevant psychological theory	Source material about relevant psychological theories and findings.
Appendix C	Interview notes	Notes from interviews with analysts involved in the study.
Appendix D	Forms used during experiment	Questionnaire forms used in the experiment.
Appendix E	Models	Finished versions of the models developed during this research work.
Bibliography	Bibliography	List of referenced publications.

Table 1.3 Chapters and appendices

2
Research Design

This chapter explains the research approach taken in this work. We shall look at the range of methods available to the information systems researcher and we shall also consider the issue of rigour and its application to the research done in this work.

2.1 Introduction

The selection of research methods must be driven by the specific circumstances and goals of each research project. This choice may be made only from an informed standpoint. Use of inappropriate methods and poorly designed research can produce misleading results, leading ultimately to inappropriate conclusions.

In this project we hope to apply psychology to improve the practice of conceptual modelling. We have already seen a range of psychological theories and results that may be of some use. But the question is how to apply these ideas in a useful and measurable way. This chapter presents a research method designed to achieve this purpose.

One starting point could be documentation about methods. Appendix A reviews conceptual modelling techniques and the information systems methods they form part of. It would be convenient if documentation of methods could tell us what we need to know about current conceptual modelling practice. But there is evidence to support the view that IS development methods are often "*a necessary fiction to present an image of control or to provide a symbolic status*" that may be "*too mechanistic to be of much use in the detailed, day-to-day organization of systems developers' activities*" (Nandhakumar and Avison 1999). It is only in practical day-to-day work that real-world problems arise. We cannot hope to gain access to current practice in conceptual modelling solely by studying documentation on, or claims made about, individual methods.

This places us in the domain of empirical research. We need to conduct research or make measurements of actual conceptual modelling practice. Such measurement could be done in a range of circumstances, artificial or 'real'. Our problem is to construct a research vehicle to allow this measurement to take place in a way that yields useful knowledge. The most useful knowledge, in this case, would be knowledge about the performance of modellers in real modelling situations, and about the impact that psychologically inspired intervention might have on their performance.

There are several barriers to gaining this knowledge. Firstly, 'real' modelling occurs in a huge range of situations, in organisations of every description, and is subject to an unbounded range of factors. There is probably no single set of conditions that could be termed 'real' modelling. And, realistically, one could not hope to conduct any kind of scientific experiment in which conditions were convincingly controlled. Results would have little generalisability, in the statistical sense, because of the uniqueness of each modelling situation.

Secondly, intervention by the researcher for the purposes of measurement cannot be avoided. It is inevitable that the introduction of psychologically inspired modelling methods will in some way disturb the situations being observed. This disturbance is likely to affect modelling performance.

Thirdly, we have many potentially useful psychological ideas, but no basis for applying them. We do not know how to introduce psychological thinking into the conceptual modelling process.

Finally, there is no established way of measuring a conceptual modeller's performance. Standards or measures of correctness do not exist. Text books on object and data modelling offer various woolly guidelines, often talking about identifying real-world concepts, but they provide no means of judging a modeller's level of success in doing this. How can we know if a modeller has modelled effectively, beyond a basic assessment of notational correctness?

While these barriers do not necessarily negate this research, we must address them meaningfully. After considering the range of research methods available to the information systems researcher, we shall return to these problems in the context of the current research study.

2.2 Research methods

Below we consider research methods according to whether they address issues qualitatively or quantitatively, how data is collected, how data is analysed, and the underlying philosophical viewpoints or beliefs about reality and the nature of research.

Questions of legitimacy and quality of research are often to the forefront in the minds of researchers (Straub 1991, Lee and Liebenau 1991), and we shall address these questions with specific reference to the information systems field. Recognising the fact that most research projects rely on a combination of methods, we shall also consider the practicalities of using different methods in conjunction with one another.

Underlying philosophical viewpoints or beliefs

At the most fundamental level, basic philosophical assumptions guide the thinking behind any research effort, whether the assumptions are stated or not. Every research study is based on assumptions about the nature of organisational reality (ontology) and about what we can know (epistemology). For example, we may assume the existence of an objective reality, or we may believe that reality is socially constructed. We may hope that our research yields universal truths, or we may feel that the results have relevance and meaning only in particular contexts. Research may be carried out using unstated assumptions on these questions, or using an explicitly stated theoretical standpoint, such as *critical social theory* (Habermas 1972, Ngwenyama 1991) or *feminism* (Stanley and Wise 1983). A researcher cannot avoid taking some position, whether consciously or not.

In general, there is nothing inherent in specific research techniques (such as data gathering methods) that ties them to one particular ontological or epistemological standpoint (Burrell and Morgan 1979). But it is common for the underlying assumptions of research to remain undeclared. And many studies are based not on a single, explicit philosophical standpoint, but on a variety of unstated assumptions.

A number of distinct and competing standpoints can be discerned in today's philosophical literature. The *objectivist* (realist) position is that a common, shared, objective reality exists and that we have access to objective reality via our senses. Truth is absolutely independent of the observer. Objectivist research (sometimes called *positivist*) seeks to establish absolute truths by measuring reality directly. The objectivist position can be thought of as an extension of the Western 'common-sense' perception of reality.

In contrast, the *constructivist* view is that reality is, for the most part, socially constructed. According to this view reality cannot have meaning independently of the observer. Reality comes about through a process of social interaction in which shared meanings are negotiated. Linked to this view is the *subjectivist* position, which holds that our experience of reality is inevitably subjective and therefore peculiar to ourselves. Some constructivists hold that there is no externally existing reality at all.

It is interesting to consider these philosophical positions from a psychological perspective. Objectivism and subjectivism are apparently contradictory philosophical viewpoints. But current thinking on cognition would tend to support both the objectivist and constructivist positions, to some extent, if we accept that our experience of the world is a result of perceptual processes. From a psychological perspective, each individual's experience of reality must be mediated by their own senses, which themselves develop as a result of the individual's unique experiences in early life. Innate developmental mechanisms normally ensure that our perceptual systems and cognitive functions develop along the right lines. But, for the most part, what we see and hear, and how we interpret what we see and hear, are unique to the individual, because the mechanisms through which we perceive and think are physically shaped by early experience. No two individuals are exactly alike. Shared location and culture tends to guarantee that individuals will have similar early experiences and hence will tend to perceive later experiences similarly. But this is not a universal rule about human beings and there can be enormous differences in the early experiences, and hence the later perceptions, of individuals raised in different locations, cultures and traditions.

The ongoing and unresolved "nature/nurture" debate highlights the further possibility that small genetic variation between individuals can give rise to larger divergences during the developmental process, leading to fundamental difference in adult perception.

The study of group psychology has shown that the opinions voiced by people in groups are often quite different from those expressed privately. From a psychological perspective, the constructivist position must hold, to some extent, since we are physically unable to experience reality directly and our thoughts and actions seem to be moulded through interactions with others. However, psychological theory would not support the extreme constructivist idea that denies the existence of any external reality at all, since current psychological theories at least are based on the idea of perceptual mechanisms that process incoming stimuli from an external world—one that is not itself shaped by social processes.

Types of data

The distinction most often observed when referring to research methods is the one between *qualitative* and *quantitative* methods. Traditionally, the distinction has been that quantitative research deals with numbers while qualitative research deals with words and ideas. When we work with numbers we can produce numerically precise and specific results, which are meaningful provided that we know exactly what we have measured and what the numbers refer to. When we work with words and ideas we are gaining a more intuitive understanding of a situation or phenomenon. Each type of data requires appropriate data collection methods (see next section).

Quantitative research is often used in the natural sciences. A major perceived advantage is that it provides reassuringly 'hard' data upon which to base conclusions. Bryman (1989) classifies quantitative data collection methods under three headings: (a) questionnaire surveys, (b) field experiments and (c) laboratory experiments. In the social sciences, organisational quantitative research experiments often take a quantitative approach, emulating the scientific method in which natural laws are established through analysis of empirical data. Perhaps because of this apparent rigour, quantitative research has generally been a preferred method for organisational studies. However, in organisational research it is often impossible to provide a truly objective, 'scientific', experiment in which independent variables are controlled or even fully appreciated. Consequently, different studies often throw up apparently contradictory results.

According to Silverman (1993) there are four major methods of *qualitative* research: (a) observation, including participant observation, (b) analysis of texts (hermeneutics), (c) interviews, and (d) recording and transcribing. In contrast with quantitative research methods, qualitative research does not offer 'hard' results. But qualitative research methods can produce results that fully reflect the richness and the subjective, interlinked, nature of organisational reality. Measurement in this type of research consists of observing and recording observations and impressions, sometimes in a rather informal way. Involvement by the researcher in the organisation is often acknowledged as inevitable. For example, in action research (Whyte 1991) intervention is seen as a key element of the research method.

But the distinction between qualitative and quantitative research methods runs deeper than the discussion above would imply. Historically, the two types of method have been used by different communities (Snow 1959). Different vocabularies are employed and there remains regrettably little mutual understanding or even common ground between the proponents of each type of method (King and Applegate 1997, Markus 1997). Quantitative research is the basis of science, and the body of theory surrounding it (statistical analysis and the theory of statistical certainty, for instance) has grown in tandem with the rise of the scientific method over the last two hundred years. Qualitative research, in contrast, sprang from the humanities and it uses techniques, such as argument and analogy, which are appropriate in the context of philosophy and the arts. Between these two extremes has sprung up a range of methods to meet the demands of disciplines that fall between hard science and the arts.

The last few years have seen a growth in both academic and industrial use of qualitative methods. For instance, the growing market research and consumer demographics industries rely on clever use of both qualitative and quantitative methods. But in some academic circles researchers who use quantitative methods still have to go to some lengths to demonstrate the validity of their methods (Cassell and Symon 1994). Quantitative methods are assumed to be 'scientific' and therefore inherently reliable and valid. Qualitative methods are regarded by some as 'unscientific' and therefore inherently less reliable and less valid.

Neither of these two views is particularly helpful since both miss the point that what is useful in one situation may not be useful in another. For instance, one would not expect to use qualitative methods in studying atomic physics. By the same token, quantitative methods would be less useful in helping to understand the reasons why software projects fail. In general, quantitative methods are particularly suited to situations where directly observable physical phenomena can be measured accurately and unambiguously (e.g. experiments in chemistry). Qualitative methods are particularly useful in explicating complex, multi-faceted social situations.

In practice, the usefulness of any particular method is governed strongly by the ingenuity of the researcher. Most quantitative researchers would admit that they indulge in quite a lot of qualitative or informal thinking while doing research. Any scientist will acknowledge the value of 'hunches', inspired guesswork and intuitive reasoning. The idea that science prohibits creative and intuitive thought is a misreading of the scientific method. However, convention dictates that only formal, analytical argument may be reported in scientific work, and informal thinking is rarely considered suitable for publication. This can be seen in those social sciences where quantitative methods have traditionally been used—probably to ensure scientific respectability rather than because of any particular suitability.

With regard to the qualitative/quantitative debate, the information systems discipline finds itself in something of a cleft stick (Markus 1997, Nissen, Klein and Hirschheim 1991). Because of its diverse origins, the field includes different groups who prefer different research methods. Information systems researchers who work in software engineering and computer science faculties typically approach research as 'research-and-development', and their preferred research method is to develop new technologies (e.g. by writing computer software). On the other hand, information systems researchers in business schools (and particularly those in North America) make great use of organisational studies with quantitative analysis, especially using surveys (King and Applegate 1997). Overall, the information systems research literature contains a good deal of quantitative, positivist, objectivist research.

Information systems researchers in Europe, and especially in Scandinavia (Olaisen 1991), have been more eclectic in their choice of methods, and in recent years, have devoted a great deal of attention to qualitative methods. To some extent this European debate has spread to the rest of the world (Lee, Liebenau and DeGross 1997) and qualitative methods are now catching up with quantitative approaches as a commonly used basis for information systems research.

It is clear that qualitative research offers the information systems researcher many opportunities that quantitative methods do not. When conducting information systems research, unless one takes an extremely narrow view of information systems, one cannot avoid looking at the *whole* situation—people, organisations, documents, processes, systems, ideas, and so on. Qualitative research methods are uniquely equipped for this purpose since they do not necessarily require the researcher to take a narrow focus and they free the researcher from the restrictions of dealing only with quantifiable measures.

Methods for collection of data

The information systems researcher may collect data in a variety of ways, whether the data itself is quantitative or qualitative.

If one is interested in obtaining statistically generalisable results from a wide population, then *questionnaires* and *surveys* may be used (Newsted, Chin, Ngwenyama and Lee 1966). However, the questions in a questionnaire or survey must be designed very carefully so that the instrument measures the desired factors effectively. It is often not a simple matter to decide which factors to measure and how to measure them. For example, do intelligence tests measure intelligence or do they measure the ability to complete intelligence tests[1]? Does intelligence have any objective existence outside the realm of intelligence tests? Questions like these must be asked of any questionnaire or survey instrument.

In addition, survey questions tend to be interpreted by respondents in a subjective way that is strongly influenced by context. Respondents do not necessarily answer truthfully and may be affected by perceived expectations. One should guard against placing a literal interpretation on questionnaire responses (Silverman 1993).

Interviews allow in-depth questioning and exploration of an individual's own perceptions and beliefs. Depending on the interviewer's intentions, an interview may be structured to a greater or lesser degree. Highly structured interviews, in which questioning follows a predetermined pattern and conversation is tightly controlled, have in the past been used in an attempt to achieve 'scientific' levels of control. On the other hand, a high degree of structure precludes opportunistic exploration of issues that come to light during the interview. Many researchers use a 'semi-structured' form of interview, in which a script is prepared in advance but deviation from the script is permitted.

An interview may be recorded on audio or video tape for later analysis (Ruhleder and Jordan 1997). It is normal practice to transcribe recordings of interviews and to work with the transcription, rather than with the recording itself, for practical reasons. Formalised 'editing' methods such as *grounded theory* (Strauss 1987) or quasi-statistical analysis techniques (King 1994) may be used in the analysis of interview transcriptions. The presence of a microphone or video camera will inevitably have some effect on the interviewee, as will the interviewer's own characteristics and the other circumstances of the interview. The questions that are posed will also have a strong influence. In any case, as in the analysis of survey data, the researcher must guard against an uncritically literal interpretation of the interviewee's words.

Verbal protocol techniques use a form of interview in which the interviewee is asked to 'think out loud' whilst performing some task (Johnson and Briggs 1994). This is a useful technique that helps to reveal mental models and thought processes. It avoids some of the problems of after-the-fact rationalisation, which can dog the conventional debriefing-style interview. Verbal protocol techniques are frequently used in research into computer user interfaces, for example. However, thinking out loud is easier for some people than others, and can interfere with carrying out the task itself.

---

[1] "*The (intelligence) scale, properly speaking, does not permit the measure of the intelligence, because intellectual qualities are not superposable and therefore cannot be measured as linear surfaces are measured*" Alfred Binet, developer of the original intelligence test, quoted in Gould (1996) p.181

More generally, *experimental research* can be used to investigate effects when one or more underlying conditions are varied. Experiments may take place in a 'laboratory' (i.e. contrived or artificial setting) or in the field, in more-or-less natural circumstances.

Simulation is an extreme form of experiment in which 'real life' is simulated in the laboratory (or perhaps in a computer program) through the use of mathematical models or in some other way. This type of research is useful, for instance, in investigating processes such as traffic flow, manufacturing processes, the weather, consumer behaviour and the financial markets. Laboratory experiments and simulation work well when the behaviour of interest can be modelled with some degree of accuracy using mechanistic models or equations.

In organisational research the situation is rather different. Experiments involving people cannot easily be placed in an artificial setting so that external factors can be controlled. People are influenced strongly by their environment and behave differently if they are transposed to new situations. On the other hand, conducting experiments with people in their 'normal' context (i.e. in working organisations) is fraught with difficulties, not least the problems of fitting a research study into the daily working of an organisation and of establishing causality in complex, multi-faceted situations (Frohlich and Pekruhl 1996).

In traditional scientific research, the decision about where to place the experiment typically depends on the extent to which independent variables can be controlled. In the social sciences, controlling independent variables is often considered less important; it is generally held that field experiments allow greater realism than laboratory experiments, and it is rarely considered possible or desirable to achieve 'scientific' levels of control (Galliers, 1992).

One recurring theme in organisational research is the search for improved efficiency. One study (Orpen 1979) compared the job satisfaction of two groups, one of which was given 'enriched' work. The aim was to determine whether altered work patterns would allow workers to gain a better sense of satisfaction from their work. The assumption, in common with that of much organisational research, is a 'functionalist' one: that increased work satisfaction leads to greater efficiency in the organisation.

According to Bryman (1989) *"a very large proportion of organisational studies are ... concerned with practical issues"*. The same author also acknowledges that *"doing research in organisations presents particular problems"*. In any research effort rigour and attention to due process are important. But the methods used in organisational studies tend to deviate somewhat from accepted norms of scientific research. For example, interpretative case studies are inevitably highly subjective, and yet are considered 'good science' (Dyer and Wilkins 1991).

"The idealised deductive process of developing a theory, deriving hypotheses, and testing them to support or not support the theory, is respected by almost everyone, but at the same time almost

everyone realises that the ideal seldom describes reality" (Campbell 1985). In other words it is difficult, if not impossible, to conduct practical research projects in organisations according to the positivist principles of the classical scientific method. Science is based on the idea of validating hypotheses through empirical observation. This approach works well where observation can yield accurate and unambiguous results, as in measurement of, say, mass. It is arguably less effective when we are measuring something whose existence may be entirely conceptual or even in doubt, such as self-esteem or job satisfaction.

Research in which the investigator learns first-hand about foreign cultures is called *ethnography* (literally, *culture-writing*) (Schwartzmann 1993). Ethnography evolved in the field of anthropology as a way of interpreting the meaning of human behaviour in different cultures. The key factor that distinguishes ethnography from realist fieldwork (Headland, Pike and Harris 1990) is its concern with cultural sense-making, a focus on *local* interpretation (that is, understanding why people do and say what they do and say from their own perspective).

It is easy enough to measure the behaviour of human beings without attempting to develop a holistic understanding of the forces that drive their behaviour. But can we reasonably draw conclusions from what people do and say, without worrying too much about their reasons? Sometimes, it is perfectly acceptable to measure behaviour without understanding 'the whole person'. For example, political polls can achieve reasonable accuracy (i.e. predictive success) without delving into the reasons why each voter states his or her particular preference. But in other situations, a holistic understanding is indispensable. We often need to understand individuals' particular worldview and motivations, in some depth. Any research that attempts to *explain* complex social phenomena must investigate individuals, their situations, their perceptions, and their interactions with other individuals.

Ethnography has for some time been recognised as an appropriate tool for the study of information systems in organisations. To take just one situation where ethnography might be useful, for example, it is certainly the case that business people and information systems specialists often seem to come from distinct cultures, unable to understand one another's viewpoints and systems of meaning (Harvey 1997). Examples abound of situations where one group seems to have difficulty comprehending the actions of the other. Business users often complain that IT people speak a strange language full of technical terms and acronyms and that they fail to understand business needs and priorities. IT specialists, for their part, frequently complain about the apparent illogicality and inconsistency of business users, as if business users constituted a homogenous group with discernable innate characteristics.

While it is plausible that IT specialists may indeed form a group with characteristics that are different from the average (and research seems to support this view), it is very unlikely that business users share many common characteristics other then a desire or need to use computer systems. So the common characteristics of business users, as perceived by IT specialists, may well have quite a lot to do with the unique character and common culture of IT specialists. It seems, therefore, that ethnography could have a good deal to offer the information systems researcher who is interested in exploring the relationships between IT specialists and their clients.

Ethnography is of course useful in many other aspects of information systems and has recently been the focus of a good deal of attention by the information systems research community (Avison and Myers 1995). The merits of *participant observation*, in which a researcher gathers data while simultaneously participating in an organisation, are increasingly being recognised (Waddington 1994). Whether in an experimental context or not, research that is part of the daily working of an organisation offers opportunities for collection of many types of data. In research of this type the presence of the researcher inevitably has some effect on the organisation. The extent of involvement may vary, from detachment (i.e. little involvement) to full immersion and participation.

Participant observation is becoming a popular method for research in information systems (Prasad 1997, Nandhakumar 1997). It affords the opportunity for a rich and intuitive understanding of complex organisational situations, something that quantitative methods signally fail to address (at least, in any formal sense). On the other hand, participant observation is an inherently subjective process that is guaranteed to affect the observed organisation in some way. Whether desirable or not (Lee 1999), this is an issue that the researcher must address.

In *action research*, intervention by the researcher is taken to its fullest extent (Whyte 1991). Action research is immersive, in the sense that the observer participates in the work of the subject organisation(s) rather than observing from a distance. A main aim of action research is to help the organisations and individuals concerned, and the researcher's impact on the organisation is seen as a key part of the research. An iterative research cycle is normally used. Subjective and interpretative reflection on the researcher's experience is used to derive useful conclusions (Baskerville and Wood-Harper 1998). Action research has been used for some years in information systems research and is slowly gaining in popularity (Lau 1997).

Case studies offer a more 'arms-length' way of developing a rich understanding of the subject matter (Janson, Guimaraes, Brown and Taillieu 1997). In a typical case study the researcher presents a considered and multi-faceted view of an organisation, project or situation, so as to help the reader develop an understanding that goes beyond the one-dimensional. Case studies often incorporate quantitative and qualitative material from diverse sources, including questionnaires, surveys, interviews and observation. They often include data derived from document analysis, since documents offer a convenient way of obtaining large quantities of source material.

Methods for analysis of data

Participant observation and case studies both yield in-depth knowledge about the specific organisations or situations that are studied. But their results cannot be generalised in the traditional manner of 'statistically valid' research. The research can inform us only about the specific circumstances in which it was conducted (in other words, it is idiographic). Because it is contextually dependent, its results cannot simply be generalised to the wider world in the manner of statistically valid scientific research. Instead, the results may be used in support of hypotheses. This is sometimes termed 'generalising to theory' (Walsham 1993, Yin 1989). We cannot make universal claims, but we can suggest possible hypotheses. This well-known form of generalisation *"can be seen as a substitute for the statistical generalisation found in quantitative studies"* (Baskerville and Wood-Harper 1998). It is a formalisation of the everyday process by which our own specific experiences lead us to draw general conclusions and to formulate beliefs (rightly or wrongly) about life. In the specific we can find the general (Solso 1998). The process of generalising to theory is a valuable one that is, in fact, an essential component of any research. Even in hard science, the initial ideas for a hypothesis must originate somewhere, and typically they come about intuitively after some period of immersion in the subject matter of the research.

Grounded theory attempts to weave theory construction and qualitative data analysis into a continuous process of 'constant comparison' (Strauss 1987). The process homes in on increasingly well-defined interpretations of the data. The idea is to base theory on empirical data, and to adjust measurement techniques in the light of the results. A three-stage approach is generally used: (a) an initial attempt to develop categories that illuminate the data; (b) an attempt to 'saturate' the categories with many appropriate cases in order to demonstrate their relevance; (c) development of the categories into more general analytic frameworks that have relevance outside the setting (Silverman 1993). The grounded theory approach has been used with some success, although one potential criticism is its apparent emphasis on theory generation over theory testing.

Grounded theory and related 'editing' techniques (Crabtree and Miller 1992) are founded in *phenomenology,* which "*seeks to understand the experiences of individual life-worlds*" (King 1994). Phenomenology could be thought of in very rough terms as the controlled use of empathy. As in ethnographic research the researcher is effectively required to enter into the world of those being studied, so that he or she can take on the same viewpoints and experience things from the same perspectives as those being studied. In phenomenological investigation one must attempt to understand each situation on its own terms, and in its own context. The idea of an objective meaning for any situation or event is rejected in favour of subjective, contextual interpretation. In research of this type the researcher must set aside or 'bracket' (and, preferably, declare) his or her own preconceptions about the subject matter being studied.

This is the *interpretevist* position (Walsham 1993). In interpretative research, subjectivity is fully acknowledged and embraced. Argument and even forms of storytelling can play an important role (Dyer and Wilkins 1991). However, the aim is not to construct arbitrary subjective justifications for particular idiosyncratic views. The intention is to make sense of the available information within its proper context. As we know from the psychology presented in Appendix B, meaning is dependent on context. To understand people it is necessary to understand the context in which they operate *as they perceive it*.

Some formalised techniques have evolved for analysing texts using both qualitative and quantitative methods (Forster 1994). *Hermeneutics* (a traditional method of analysing classical texts) has been applied as a method for the study of organisations (Boland 1991) in which the organisation is interpreted in a similar way to a text.

Moving to positivist research methods, *statistical theory* offers many ways of analysing quantitative data. For example, the correlation of two sets of measurements may be calculated to determine the extent to which they are related to one another. Standard distribution patterns, such as the Normal and Poisson distributions, are used to determine whether numerical results deviate from an expected range of values. Confidence levels may be calculated for numerical results, indicating the degree of certainty that one may place on the results.

Use of statistical theory requires control of independent variables and depends heavily on objective measurement. Consequently, it is often used in scientific studies such as laboratory experiments where a high degree of control and accuracy can be achieved. It is also used in the social sciences to analyse data arising from questionnaires and surveys. In these situations, statistical theory must be applied with great caution since its results rely on an absence of bias (i.e. on objective measurement), something that is not always easy to guarantee when humans answer questions or report impressions.

It is often the case that different methods are used together in a single research study. When the results from one method or study are used to check the results from another method or study, this is termed *triangulation* (Denzin 1970, Cassell and Symon 1994). The benefit of triangulation is that it can provide some perspective on the results of research. The flaws (e.g. bias) in one study are unlikely to coincide with the flaws in another study conducted using different research techniques. Comparison of results between the two can offer some indication of accuracy, either corroborating or confounding the results.

The methods used in triangulation may well spring from different philosophical viewpoints (e.g. positivist survey research triangulated with interpretevist action research). There is no inherent contradiction if they do, although one should be careful to ensure that one is comparing like with like, since the material of positivist research (objective measurement) is often rather different from that of interpretevist research (subjective meanings). Silverman (1993) goes so far as to reject triangulation outright as an aid in adjudicating between alternative accounts largely because of this risk.

Research methods in the social sciences

The scientific method depends on the researcher's ability to postulate causal mechanisms that can be tested through observation. Without a theory, we have nothing to test. The results of this process are working hypotheses. Examples of such hypotheses include Newton's theory of gravitation, the theory of quantum mechanics and Darwin's theory of natural selection. These theories are not *facts*, although many scientists and non-scientists take them as such. There is a widespread perception that science claims to provide access to universal truth. But this perception is mistaken.

In fact, science relies more on disproof than on proof. It is rarely possible to prove that a theory is universally true, only to demonstrate that it is not. Hence any theory must be capable of being refuted. A theory that cannot be disproved is ultimately of little value in this respect except, perhaps, to stimulate debate. Research that cannot credibly be relied on to disprove invalid theories tends to be dismissed as pseudo-science or speculation. In the realm of psychology, Freud's theory of the Oedipal complex and Schank's theory of scripts are examples of theories that cannot be disproved.

There is a huge range of research problems, therefore, where the classical scientific method is of little use. Much of psychology, sociology, anthropology, management and other disciplines cannot use this method. Yet in these fields research methods have regularly been employed that mimic the scientific method. Arguably, pseudo-scientific research has been used in the social sciences because of the inherent respectability attached to the scientific method. Scientific forms of rigour are rarely attainable in these disciplines. But credibility is lent to research that is couched in the terminology of science and whose results are presented in a way that apes the results of scientific experiments.

The rush for 'scientific' forms of rigour in the social sciences is unfortunate, not least because situations in which people are involved offer a rich and fruitful vein of data for the researcher to mine. Exclusive use of quantitative methods, with statistical significance testing, leads to a poverty of ideas and a diminished understanding of the situations being studied.

It is illuminating to consider this situation in the light of what we now know about the workings of the brain. A belief in the primacy of rationality and the will (that is, of conscious thought and logic) is still a popular philosophical position. But this position is based on a view of the mind that is coming under increasing threat as more is discovered about how the brain operates. Connectionist research indicates that most of intelligence is unconscious and intuitive, and not the product of conscious, analytical thought (Solso 1998). If unconscious mental processing forms the basis for most of our thinking, it must therefore be behind most scientific achievements. But research methods that explicitly acknowledge the application of unconscious or intuitive thought are generally excluded as 'unscientific'.

A characteristic of all research is that, whatever techniques and philosophies are used, the researcher inevitably ends up rationalising about the results. Logic and rational argument are used to argue in favour of particular conclusions. As long as we continue to use language, this will be true regardless of whatever frameworks or philosophical standpoints have been used to guide the research. But rationalisation is derived from unconscious thought. Inevitably, our own rationalisations are inaccurate since they verbalise only a fragment of the full life experience behind any conclusion. We are able to draw conclusions, but we are rarely able to pinpoint with any accuracy precisely what factors contribute to our conclusions. It is very difficult to separate interpretation from research, and probably fruitless to try.

Summary: Research "dimensions"

As we have noted, the means by which research is carried out can be classified in a number of ways. Researchers may use qualitative or quantitative data-gathering methods. They may presume that an objective reality is being observed or they may take the view that reality can be experienced only subjectively. They may adopt a critical stance or try to be neutral. They may remain detached or they may become participants in the observed organisations. Table 2.1 presents some of these dimensions.

Structure of project	**Involvement of researcher**	**Analysis**	**Data sources**
Lab. experiment Field experiment Case study Simulation Survey Grounded theory Triangulation	Detached Immersive (e.g. ethnography) Participant observer (e.g. action research)	Statistical Graphical Narrative Semiotic Hermeneutic Metaphoric	Interviews Questionnaires Direct observation Measuring Document analysis
Interpretation of data (epistemology)	**View of data (ontology)**	**Data**	**Generalisability**
Neutral (positivist) Interpretative Critical	Realist (objective reality) Nominalist (subjective reality)	Quantitative Qualitative	Wide (nomothetic) Narrow (ideographic)

Table 2.1 Summary of research dimensions

Eight separate dimensions are used in Table 2.1, implying that any research study can be placed at a specific position in an 8-dimensional "research space". There is a tendency to treat the alternatives within each dimension as mutually exclusive categories, so that any given piece of research has to be placed into one or other category. But, in practice, a mixed approach is quite viable (Trauth and O'Connor 1991). For example, a research project can involve both qualitative and quantitative data gathering. It can acknowledge the subjective nature of reality but treat it as if it were objective for some purposes. The researcher may be engaged at one point, detached at another.

2.3 Information systems research

The field of information systems is relatively new in comparison with many other academic disciplines. It is a discipline that incorporates a number of alternative viewpoints (Powell 1998). Information systems academics are found in computer science, engineering, statistics, business and social science departments. Information systems impinge on almost every aspect of modern life and consequently the study of information systems encompasses a broad range of topics. It is a truly multi-disciplinary subject area. Elements of sociology, psychology, information science, software engineering, management, operations research, organisational behaviour, economics, finance and mathematics combine to make up a single discipline concerned with the use of information and information technology in organisations (Avison and Myers 1995).

A significant proportion of the research in information systems to date has used positivist and objectivist survey and experimental methods, often with quantitative analysis (96.8% according to one survey) (Orlikowski and Baroudi 1991). Arguably, there have been strong pressures for methods of this type to be used in information systems research, rather than alternatives such as qualitative methods which may be more suitable (King and Applegate 1997).

One stream of information systems research has been influenced heavily by technological advances in the computer industry. This research community has tended to focus on 'information system as software system' rather than 'information system as social system'. Examples of this type of research include the work during the 1980s of IFIP Working Group 8.1 on system development methods (Olle et al 1982) and, more recently, the many semi-academic conferences on object-orientation (Briggs and Werth 1994).

In recent years a pluralist position has begun to evolve, and debate has arisen about the suitability of a wide range of research methods (Lee, Liebenau and DeGross 1997). It is argued that information systems are social systems and hence less amenable to positivist methods, positivism being most appropriate in the physical sciences where conditions can be tightly controlled and objective measurement is feasible. In information systems a more subjective, qualitative research approach is called for. It is hard to argue with this position since work in or with information systems often involves people, organisations, customs, office politics, legislation, information and technology as well as a range of other concerns. Moreover, the process of information systems development is often itself rather politicised, with various stakeholders who may have competing interests. 'Truth' is often a moveable feast. Quite understandably, qualitative research methods are now discussed and used with increasing frequency in information systems research.

Underlying this debate is a move away from a concentration on computer-related aspects and towards wider business issues. The term 'information system' itself embodies this move since it is generally considered to include *all* aspects of organisational systems concerned with the manipulation and delivery of information, including people and what they do.

The debate about research methods in information systems has perhaps matured to the stage where it is no longer dominated by a reaction against positivist methods but has begun instead to embrace methodological pluralism. It seems that the information systems research community now feels sufficient self-confidence to be able to choose whatever methods are appropriate for each piece of research, without fear of their chosen methods being branded as lacking in rigour or validity.

One sign of this maturity and confidence is the fact that experienced information systems researchers (e.g. Markus 1997) are bemoaning the dearth of *practical* research that employs qualitative methods but integrates both technological and social aspects. For instance, there has historically been a genuine dearth of research in which new design techniques are seriously validated (Fitzgerald 1991). The present work is an attempt at this type of research.

Rigour in information systems research

Many information systems researchers have addressed the issue of rigour in information systems research (e.g. Keen 1991, Turner 1991, Galliers 1994, Avison and Nandhakumar 1995, Lee 1999). For any research to be acceptable to the information systems research community it must be rigorous and be seen as rigorous. The definition of rigour depends on circumstances. A piece of research is likely to be judged to be rigorous if its methods are justifiable and have been applied in accordance with established best practice *in its own particular field.* For historical and other reasons, different research methods are often considered appropriate in different disciplines, even when common subject matter is being studied (Lee 1999). This means, of course, that some disagreement may emerge when cross-disciplinary research is attempted or when research methods are "borrowed" from reference disciplines.

In general, qualitative information systems research must be conducted in a transparent, open and even-handed fashion. If assumptions are made, they must be stated. Pre-conceived ideas must be declared and "bracketed" (placed to one side). This requires a degree of honesty and self-examination on the part of the researcher. In cases where the research involves interpretation (e.g. when the significance of variation between several data sources needs to be evaluated), the researcher must make plain what reasoning and what evidence has been used.

In quantitative research, rigour often means experimental objectivity and control, with due attention being paid to management of experimental conditions and ensuring that measurement is conducted accurately. Accuracy is usually judged in two ways: *reliability* and *validity*. Reliability refers to the consistency with which a given phenomenon is measured. A research measure is reliable if it consistently yields the same (or similar) result when measuring a particular property. Validity refers to the degree to which a research measure corresponds to the real-world property of interest. A research measure is valid if it is a good indicator of the concept in question.

A simple example serves to illustrate the difference between these two measures. A thermometer that consistently yields the same reading of 82°C when placed in boiling water is reliable but not valid. A thermometer that gives varying readings centred around 100°C is valid but not particularly reliable (Kirk and Miller 1986).

It is possible to distinguish two aspects of reliability: *internal* reliability and *external* reliability. External reliability indicates the degree to which a measure is consistent *over time*. Internal reliability refers to the degree to which a measure is *internally consistent* (Bryman 1989). This issue is relevant in the case where several 'facets' or measures are combined to form a single 'dimension'. To measure internal reliability, we can calculate the average correlation between the facets comprising the dimension. For example, if we combine separately determined measures of absenteeism, self-esteem, productivity, and so on, into a single factor 'job satisfaction' then we must calculate the internal reliability of this measure. Normally a correlation of .8 or greater is considered to be acceptable (Bryman 1989).

Several aspects of validity may be distinguished. A measure has *face* validity if it appears at first sight to be representative of the property of interest. Other types of validity include *criterion* validity and *construct* validity, both of which seek to link the thing being measured to the concept of interest in understandable ways. To produce a valid outcome it makes sense to measure something that is closely related to the concept of interest. In an ideal situation, the property of interest can be measured directly. For example, a direct measure of job satisfaction would be demonstrably more valid than measuring absenteeism, physiological stress levels, or even employees' assertions about their own job satisfaction. But more often than not, the property of interest is an abstraction that is incapable of direct measurement (such as job satisfaction itself).

The requirement for validity means that we have to be very cautious when interpreting research data. For example, we must be careful to interpret questionnaire responses in a valid way—as *assessments by the user* of something. We cannot take them at face value. Questionnaire responses are not necessarily reliable indicators of fact or even of opinion since participants may misreport what they think for one reason or another. For example, it is easy for someone to believe they understand something when they do not. One may consciously or unconsciously misreport one's own level of knowledge for a variety of reasons, including peer pressure and a desire to avoid appearing foolish. It has long been recognised that the self-report questionnaire should be augmented by other data sources if accurate data is to be obtained on organisational phenomena (Campbell and Fiske 1959).

Similarly, when conducting interviews it is not sufficient to report the responses given by subjects as if they were simply an account of an externally existing reality. As Silverman (1993) observes, interview data can be interpreted as a 'situated account' in which the informant constructs a version of reality which they feel is appropriate, given the circumstances of the interview. Bryman (1989) refers to problems of this sort as 'reactivity' and 'social desirability bias' and observes that there is often *"a gap between ... what people say they feel and how they actually feel"*. The way in which questions are framed can also influence the interviewee's attitudes and responses. This can be due to factors such as cultural differences, inhibition and the interviewer's own mental framework or internal coding scheme. The interviewer's unconscious preconceptions can form a *"powerful conceptual grid"* that is hard to escape from (Atkinson 1992). When one's motives or reasons for a particular course of action or view are questioned, it is all too easy to produce plausible rationalisations in the absence of actual self-awareness or honesty, especially if the truth might be painful or embarrassing. Consequently one must guard against simple-minded or literal interpretation of interview responses.

In quantitative research, validity and reliability are especially important. It can be all too easy to resort to anecdotal evidence or to generalise from single cases. Silverman (1993) offers several means of ensuring reliability and validity in qualitative research when different research methods are used:

Research involving *participant observation* often involves the reproduction of selected observations in support of the researcher's own hypotheses. The reader may have no opportunity to review the source transcripts from which the selected items have been taken. Several researchers have argued that rigour in this situation demands 'openness', with all source materials being made available (Spradley 1979). For example, original notes and transcripts should be provided, together with preliminary attempts at coding and analysis. The reader can then judge more easily the credibility of the researcher's interpretations. A second way in which openness can be achieved is by using a formal notation to distinguish between (a) interpretation and comments from the researcher's point of view and (b) analysis that springs from the conceptual framework of those being studied (Kirk and Miller 1986).

When *analysing texts*, some of the transcription-related problems that derive from the use of notes and observations are avoided since the text is already available. However, to ensure reliability it is important that a consistent set of analysis categories be used, and the categories should be applied in a standardised fashion. A good way of determining whether textual analysis is reliable is by assessing 'inter-rater reliability' (Silverman 1993) in which several assessors use the same categories to analyse a single set of data.

When conducting research *interviews* it is common to try to ensure reliability by pre-testing interview schedules, training interviewers, using fixed-choice answers and using inter-rater reliability checks. Although these are often worthwhile precautions, Silverman (1993) points out that concentrating on these issues may obscure the problem of attaching literal meaning to interviewees' answers. As we have already noted, from the point of view of validity, interviewees' statements cannot usually be taken as direct statements of fact: "*What people say in answer to interview questions does not have a stable relationship to how they behave in naturally-occurring situations*" (Silverman 1993 p150).

Similar problems exist in formulating questionnaires and interpreting the responses to them. Context plays a pivotal role in interpretation. The meaning attached by each respondent to particular questions will almost inevitably differ from that intended, often in unpredictable ways. The authors of questionnaires are generally powerless to control (or even be aware of) the context within which each respondent completes the questionnaire.

One common way of analysing the data arising from questionnaires is by looking for correlation between factors. This involves the use of simple statistical techniques and can reveal interesting and otherwise hidden patterns in the data. But when analysing any data it is important to avoid the *causation fallacy*. Factors that are statistically correlated do not necessarily have a causal connection. For example, the incidence of false teeth and the incidence of arthritis in the general population are probably correlated. But it would be quite wrong to take this correlation as evidence that having false teeth causes arthritis (or vice versa!). It is easy to make this mistake. Indeed, there is psychological evidence that the human mind naturally and unconsciously perceives a causal relationship between events that occur in quick succession, for example, regardless of whether any causal relationship actually exists.

This property of the mind has probably been useful to humanity in evolutionary terms. But the tendency of researchers to perceive causation when correlation is observed is dangerous. In fact, the best we can say about any correlation between variables is that it may hint at possible causal relationships. Silverman refers to this kind of invalid assumption of causality as "*spurious correlation*", which he distinguishes from "*nonspurious*" (i.e. non-causal) correlation (1993).

In *case study* research and in research that involves qualitative analysis of small samples 'representativeness' is a common concern. Hammersley (1992) suggests that we can help to ensure that case studies are representative by consciously comparing our cases with relevant aspects of the wider population. This could be achieved, for instance, by carrying out survey research on a random sample of cases. Alternatively, a number of 'parallel' cases could be analysed. The problem can be negated altogether using the logic of 'generalising to theory', which avoids the validity-related problems of generalising to a larger population.

Conclusions

The existence of rigour in information systems research hinges on attention to validity and reliability, just as it does in the natural sciences. But it also relies on what might be termed *due process*: conducting research according to established rules and norms. The established rules and norms vary according to discipline and research method. For example, best practice in ethnographic research requires a different set of rules and norms from best practice in quantitative survey-based research.

In information systems research it is sometimes unclear exactly which rules and norms apply. Information systems could be characterised as an academic discipline caught between one set of rules and norms and another. In the past, information systems professionals and researchers could have been accused of holding an excessively mechanistic (i.e. positivistic) view of the organisation and of user needs: the "data plumbing" view of information systems. Even the methods used by information systems practitioners tend to mimic the internal workings of computer systems, reflecting a mechanistic viewpoint (e.g. the common practice of splitting the analysis of an organisation into data and process analysis, which mimics the split between data and programs in computer software).

This rather limited and naive view of the organisation (and consequently of the role of the information systems professional) persists widely in industry, and is actively promoted in many software engineering and computer science courses, which characterise the development of information systems as primarily an engineering activity. It effectively ignores the subjective nature of information systems and the roles of psychology, politics and social processes in information systems practice. Perhaps it stems from the nature of IT specialists, who as a group are known to be typically more introverted, more logical and with lower social needs than the general population. Alternatively it may be a consequence of the widespread perception that IT is chiefly a technical specialism.

Whatever the cause, it is clear that information systems *researchers* (at least) have in recent years woken up to the inadequacies of this view and now embrace subjectivity and qualitative methods generally, together with the rules, norms, and standards of rigour that apply to them. Information is a subjective thing; one person's information is another person's noise. Those who think of information as something with objective existence are failing to observe the distinction between *information* and *data*.

We are even seeing these methods gaining some small foothold in practice, thanks perhaps to the enlightened stance of HCI (human-computer interaction) practitioners. Interestingly, the HCI community, which is separate from information systems as a discipline but shares many common concerns, has long been persuaded of the need for ethnographic and other qualitative methods in both research and practice, and it uses appropriate criteria when assessing experimental rigour. The HCI field developed independently of information systems and, perhaps because of its explicit concern with the psychology of the end user, it did not start from a wholly positivist position. Consequently, HCI as an academic discipline seems to have missed out on much of the debate concerning research methods that has occupied the information systems research community in recent years.

This research study involves some elements of both the 'old' and 'new' information systems worldviews. Having been rather eclectic in our choice of methods, we are therefore obliged to address the question of rigour independently for each research technique. In section 2.4 we present the research techniques that were selected for use in this project and in section 2.5 a step-by-step analysis is given of the ways in which these techniques were used. For each step we return to the issue of rigour, showing how action was taken to ensure that best practice was observed in the context of each distinct research technique.

2.4 Selecting appropriate methods

This section presents the methods selected for use in the research project. In reality, the research method was revised several times, as the project progressed, before taking its final shape. It is probably common for research methods to evolve in long-term projects such as this one. Within the conventions of a doctoral thesis it is difficult, however, to present a research method in a way that reflects its evolutionary nature. Therefore only the end result of the process is described below.

Refining the research question

The overall aim is to determine whether the application of contemporary psychological thinking can assist in conceptual modelling. This is not a well-defined goal; there are many ways in which one might envisage psychological ideas being introduced. The enormous range of psychological theory that could be applied makes it necessary to be selective. We must choose a particular way in which selected psychological ideas can be applied to conceptual modelling, and we must also decide how the consequences can be assessed. Our key decisions are therefore:

a. Which psychology to introduce?
b. How to introduce it?
c. How to assess the consequences?

One might also ask what aspects of conceptual modelling are of interest, since conceptual modelling is itself potentially a very broad field of investigation. However, that question may be answered simply, by reference to the definitions of conceptual modelling given in Chapter 1. There we gave two definitions of conceptual modelling, one broad and the other narrow. In the experimental portion of the research it is the narrower of the two definitions that will be used. The narrow definition sees conceptual modelling as equivalent to UML-style object modelling, entity-relationship modelling and similar object and data modelling techniques. The experiment will adhere to the *narrow* definition of conceptual modelling and investigate the impact of psychologically inspired thinking on techniques of this sort.

Because these modelling techniques are closely related to one another, it is sufficient for testing purposes to use simple object/data modelling notation that encapsulate the main features found in these techniques. The features are (using terminology from both object modelling and data modelling): classes (entity types), associations (relationships), inheritance (subtyping) and properties (attributes). More advanced features that are perhaps more typical of the expert modeller and may be peculiar to one or other style of technique (such as *multiple inheritance* and *aggregation* as used in object-oriented design) will be avoided. The techniques to be used are explained in more detail in Chapter 4.

There is ample justification for using such a "lowest-common denominator" modelling technique; it is representative of the techniques that are generally used by most non-expert analysts and designers. It is also similar to the elementary techniques taught in most introductory-level courses in object modelling and data modelling. Few people involved in designing systems go beyond this basic style of modelling to use more sophisticated constructs. Secondly, many of the more advanced constructs (such as *methods* or *operations* in object modelling) are arguably implementation–specific details. We might prefer to omit most of them from business-oriented conceptual models in any case. And, finally, models that have been constructed using the simple techniques can be every bit as 'correct' as those that use more advanced notations and constructs, since practice over many years has demonstrated that techniques of this sort can represent the bulk of business situations adequately.

We shall now deal in turn with each of the three key decisions listed above.

Which psychology to introduce?

The question of which psychological ideas to use can be addressed in a range of ways. At one extreme, we could attempt systematically to appraise all of current psychological thinking and glean from it a body of potentially relevant knowledge to apply (one might call this the "non-selective" approach). On the other hand, this research was driven (initially, at least) by some preconceived ideas about the potential relevance of particular psychological developments, including neural networks and the idea of analogical mental models. These ideas probably should not be taken in isolation, but they form an 'inspirational core' to this work. Basing research on such a narrow range of preconceived ideas could be termed the "selective" approach.

In practice, the direction taken in this work is a combination of the selective and non-selective approaches. A reasonably broad trawl through the mainstream psychological literature was made (Appendix B). This highlighted a range of ideas that could potentially yield useful results when applied to conceptual modelling. For example, ideas from group psychology, which did not form part of the initial thinking, are included in the appendix. All of these ideas are taken forward for later use. But at the same time, the original motivation behind the project is not lost and the 'inspirational core' of ideas remain as contenders for later inclusion in experimental work, provided there is reasonable evidence in the psychological literature to support their inclusion (which is also given in Appendix B).

How to introduce the psychological ideas?

The acid test for a method (and for a conceptual modelling technique in particular) is whether it is useful *when put into practice* (that is, unless it is intended simply to stimulate debate). 'In practice' in this context must mean 'when applied within organisations' since that is where conceptual modelling techniques are normally used.

There is no shortage of academic writing that presents methods which are *theoretically* useful. New methods are continually being developed and proposed. Attempts to categorise or to make sense of the huge range of methods (or "methodologies") often run into trouble because of the sheer number of methods and range of ideologies, perspectives and approaches (Olle 1991, Jayaratna 1994). One could see the continuing development of "new and improved" information systems development methods as evidence of a healthy debate and an enriching turnover of ideas. It certainly suggests that we have yet to find universally accepted solutions to the problem of designing and developing business systems.

But it is rare to see evidence that particular methods are workable in real organisations, with all of the complex, problematic reality that real organisations entail. Indeed, perhaps a useful way to cut through the plethora of methods is simply to look for those with actual evidence of practical utility. For these reasons we have chosen to base the experimental research on an investigation of conceptual modelling in the context of real business organisations. Practicality dictates that we can use only a small number of organisations and only a limited number of modellers and business end users.

By conducting field experiments with modellers and business end users who are working in their own organisations we hope to gain first-hand insight into a "life-like" form of conceptual modelling practice—as distinct from textbook practice, or the theoretical practice that the inventors of information systems methods talk about. For various reasons, real organisations often create sub-optimal conditions for modelling. In practice, real modellers are not always as well trained as they might be. Interpersonal skills and other factors can seriously affect the outcome of analysis and design efforts. Intractable issues like politics and commercial imperatives frequently have a major impact on the system development process. These issues are rarely covered in the textbooks on mainstream conceptual modelling methods, although they are addressed by some non-mainstream methods.

We are aiming to observe conceptual modelling in a realistic setting. One must not fall into the trap of assuming that one can have objective access to "real life", or that looking at "real life" will give access to universal truths. Yet it is surely fruitful to observe it as closely as possible if the opportunity is available. By choosing to look only at a small number of organisations and modellers we create the opportunity to view and analyse modelling practice at a more detailed level than would otherwise be possible, to gain a richer appreciation of what happens when modellers and business end users set out to model their businesses and design information systems.

From a philosophical point of view, our need to work with real organisations and real modellers makes it less appropriate to take a positivist or objectivist position. We cannot hope to measure modelling performance absolutely, for example. If we judge that a particular method is perhaps more useful than another method in a particular organisation, we nonetheless cannot generalise this result to *all* organisations. Increasingly in the discipline of information systems, a constructivist and interpretive stance is felt to be most appropriate in research involving people and organisations. That is not to say that all similar studies reject positivism; some quantitative and positivist studies have been reported of information systems methods in action (Munro and Davis 1977, Mumford and McDonald 1989). But, perhaps reflecting shifting ideas in the information systems community about research methods, recent studies of information systems methods have more typically adopted a qualitative approach, often using comparative methods and feature analysis (Garcia and Quek 1997).

What alternative ways of applying psychological ideas are available? Given that we plan to use a small number of modellers drawn from real organisations, several broad alternatives are possible. One approach would be to educate the modellers in psychological ideas, and then to ask them to apply the ideas they have learned, observing the results. Alternatively we could attempt to formalise the psychology in some form of predetermined procedure, which the modellers could use without necessarily knowing what psychological thinking they were applying. Or we could simply let the modellers do their work and later analyse what happened from a psychological perspective.

Clearly each of the three alternatives given above would yield an interesting research project. In fact, we have opted in this project for a combination of the second and third approaches. We shall use a way of working that conforms to certain stated psychological ideas. But we shall also allow some modellers to use traditional methods so that we can analyse and compare the results later.

Why this approach? The answer is probably that it is the one that seems most likely to yield useful results. Firstly, it permits comparison between modified and unmodified modelling methods. Hence we have some basis for assessing the difference between applying psychology and not applying psychology. Secondly, it avoids the hit-and-miss nature of the first option. If we simply educated the modellers in psychological ideas and then asked them to apply those ideas in their work, we would have scant means of assessing the uptake of those ideas or even of judging the extent to which modellers applied them.

The choice is akin to that made by researchers when deciding whether to conduct interviews in a structured, semi-structured or unstructured fashion. An unstructured interview may yield interesting and unexpected information. But it is difficult to integrate that information into a larger body of knowledge or theory because of its fragmentary nature, and because no consistent principles or framework were used that would allow the results of one interview to be compared in a convenient way against the results of another. Hence many researchers opt for semi-structured interviews in which an overly rigid and controlling approach is avoided but some framework is nevertheless applied to guide the conversation.

Therefore we chose to use a "semi-structured" approach to introducing psychology into modelling. A framework was applied by using a simple modelling method that conforms to the psychological principles. Although constrained by this framework, the modellers were otherwise essentially free to act according to their own impulses. The fact that they used a predefined method allowed us to have some basis for comparing one modeller against another, and helped to ensure that psychology was introduced in a reasonably consistent way. But because the modellers were free to create whatever model content they saw fit, we were still able to gain useful and unexpected insights into their thought processes and actions.

We chose to construct the framework in two stages. The first stage consists of a number of 'psychological principles of modelling', based on the evidence set out in Appendix B. The principles and the thinking behind them are given in Chapter 3. We then used these principles to explain and rationalise about a simple conceptual modelling technique, according to the constraints mentioned above. The conceptual modelling technique could take many forms, depending on how one interprets (and chooses) the psychological evidence. But we justify its particular form using an argument based on explicit and defensible reasoning. The technique and the justification for its particular form are both given in Chapter 3.

How to assess the consequences?

A field experiment was carried out to test the results when the simple conceptual modelling technique was used in practice. As explained above, this took place in the context of actual business organisations and used "real" modellers (i.e. real business people who for one reason or another were involved in the modelling efforts), so as to achieve some degree of realism. We were aiming to capture as closely as possible the authentic experience of a modeller having to use a modelling technique on a business problem. We used both a modified method incorporating psychological principles and an unmodified 'standard' method, to observe the results in both situations and to affect a comparison between them. Various research techniques were used during the experiment, including questionnaires, interviews, observation (including participant observation), and simple numerical analysis. These are discussed briefly below.

Questionnaires During and after their work, modellers and business users involved in modelling sessions were asked to complete a questionnaire giving their subjective impressions on different aspects of the process. The number of individuals participating in the exercise was necessarily limited by the chosen design for the research, as discussed above. This means that the resulting sample of questionnaire responses is not statistically significant. But a qualitative analysis of the responses can nevertheless provide useful insight.

Interviews A modeller who took part in many of the modelling sessions was interviewed after the modelling was complete. A semi-structured interviewing process was used, yielding useful insights into the individual's own perspective on aspects of the modelling task and technique. The advantage of interviewing in this situation was that it allowed specific episodes or issues that arose during modelling to be explored in more detail. The interview transcript was used during the later interpretive analysis.

Observation An investigator observed the proceedings in some modelling sessions. Selected sessions were recorded on videotape so that detailed analysis could be performed later. The aim was to get a clear idea of the interplay between the modeller and the business users taking part in the session, and to provide some reference points against which questionnaire responses could be compared. The modellers' use of the modelling tools was also of interest.

Participant observation An investigator worked with modellers (see below) who were working individually. Whilst not taking an active role in modelling, the investigator provided assistance in a controlled fashion and actively observed the behaviour of the modellers, keeping a record of what they said and did for later analysis while attempting to understand the underlying perceptions and motivations guiding their behaviour. The aim was to understand the modellers' actions *from their own perspective.*

Numerical analysis Although numerical analysis may seem to be a quantitative technique, it is frequently used in the context of qualitative research, where small samples are often involved. It can yield useful insights even if the results are not significant according to conventional statistical theory. In this instance the models produced in all modelling sessions were analysed and deconstructed to reveal underlying assumptions, conceptual structures and patterns of development. Various low-level measures (such as correctness, completeness, productivity, and so on) were combined to compute overall measures of modeller effectiveness and usability (i.e. usability of modelling technique, as explained in the following section).

Conceptual modelling can be a group activity or an individual one, depending on circumstances such as business needs and the capabilities or preferences of the modeller. Since we were dealing with "real" modellers, they varied in levels of ability and training, and the group overall included both non-expert and expert modellers. It is unreasonable to expect non-expert modellers to conduct group modelling sessions, which they must lead and control (as in a JAD workshop) since they are likely to lack the essential skills. However, for expert modellers it is normal and desirable to work in a group setting. Therefore we used two types of modelling session in our study: individual modelling sessions conducted by lone modellers who were less experienced, and group modelling sessions conducted by a very experienced modeller with small groups of business users.

This structural difference between individual and group modelling sessions makes necessary a diversification of research methods for each type of session. For *group* modelling sessions questionnaires were used to elicit subjective impressions from the modeller and business users who took part. Interviews were used for a more in-depth exploration of specific issues. Observation was used to discover how the modeller and business users went about the process of modelling. Simple numerical analysis was applied to the models that were produced. In *individual* modelling sessions an investigator acted as participant observer, to help guide the less experienced modellers and to gain a closer understanding of individual modellers' actions and motivations. Because of this, questionnaires and interviews were not used. Simple numerical analysis was applied to the models produced in the sessions, as for group sessions.

Conclusion: Overall structure

The chosen research design for this project was divided into two phases: *theory development* and *theory testing*. The theory development part culminated in the formulation of modelling principles and a discussion of their relevance to a modified modelling technique, as outlined above. This stage could be characterised as "conceptual study" (Avison and Myers 1995). The theory testing part of the research included a field experiment in which use of the new modelling technique was studied.

The research design is shown in Figure 2.1. In the figure the theory development phase is divided into three stages, starting with "reflective practice" and continuing with a literature survey before culminating in the formulation of principles and method. This division reflects the genesis of the research in practice-based work, and the subsequent evolution of the ideas behind it. The theory testing stage is divided into the field experiment described above, which produced results in various forms, and subsequent stages in which the results were interpreted and final conclusions drawn.

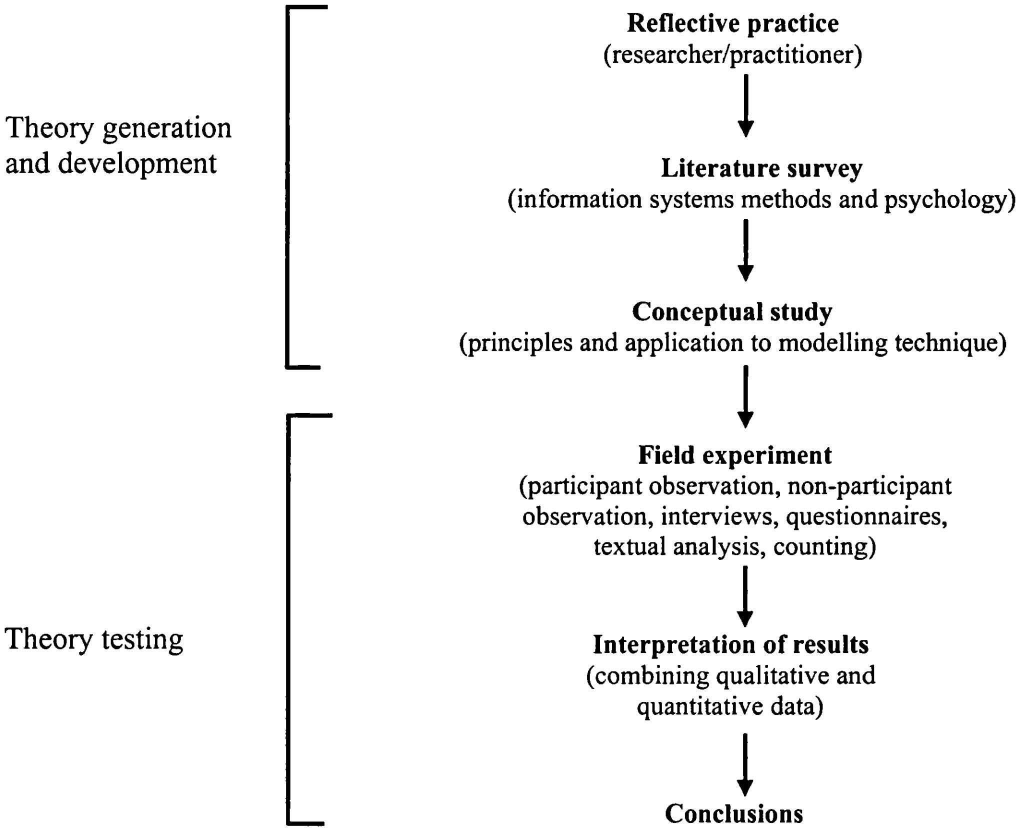


Figure 2.1 Chosen research design

Although this overall structure is frequently used in research projects (Scott Morton 1984), the chosen research design in this case is quite heavy on theory development. One reason for the emphasis on theory generation is that conceptual modelling is not a 'green-field' research area. The project requires the development and expression of new thinking in a field that is already saturated with ideas, many of which have become so well established as to be regarded as fact. Some of the ideas in this thesis are perhaps contrary to current received wisdom and therefore require full and careful exposition.

Overall, the research could be characterised primarily as qualitative research. Part of it (the field experiment) involves quantification of various measures and some numerical analysis. But the results of the analysis are interpreted in a qualitative manner. Specific figures are of less interest than the 'shape' of the results and the ways in which they add significance to the subjective impressions and other data gathered during the project.

2.5 Selected methods in detail

The following sections give a detailed account of each stage of the research, focusing on the methods that were applied and the ways in which rigour and best practice were observed. Note that we concentrate here on research method. The results of the research are given in Chapters 3-7 of this thesis.

Theory generation and development - Reflective practice

The overall aim of the research (see Chapter 1) was arrived at after a long period of teaching, research, practice, study and reflection. This involved practice of project management and systems analysis and design (including conceptual modelling) in commercial organisations in parallel with teaching of the same subjects at third-level institutions (McGinnes 1994b). The research work included comparative analyses of system development methods (McGinnes 1992, 1993, 1994a) and investigations into neural networks (McGinnes 1991). The study encompassed a range of contemporary topics in psychology, information systems and computer science at postgraduate level.

Having to teach a subject that one practices is a natural encouragement to reflect upon experience. To teach a subject one must think about it. And in practical subjects like software project management, systems analysis, and conceptual modelling it is natural to turn to one's own experience for inspiration in teaching. Initially, the researcher's personal experiences were used as a source for teaching in case studies and as anecdotal material to add colour to lectures. But as the teaching cycle repeated itself, and research got under way, new perspectives on past and present experiences in consulting practice began to emerge. Teaching and research led to the development of mental frameworks that naturally became useful in thinking about practical work.

This period could be characterised as one of *reflective practice* (Schön 1983). It engendered a developing sense of frustration with the conventional ways in which software professionals go about helping their customers. It became increasingly apparent that current practice is not necessarily built on a sound theoretical base. Even the "ideal" ways of working put forward by leading IT industry figures—which many professionals aspire to—seemed to lack a genuine foundation in the relevant areas of knowledge (e.g. cognitive psychology and group dynamics). They seemed to be derived more from logical development and refinement of earlier ways of working. After ten years in the information technology industry, with little exposure to formal or theoretical underpinnings, an introduction to the academic world led the researcher to develop new mental frameworks and views. One new insight was the idea that learning "about" and learning "how" are not the same thing. The ability to recall facts about a thing or situation is not the same skill as being able to use the thing properly or act appropriately in the situation. To a new entrant to third-level education, it seemed that university academics tended to concentrate on getting students to know "about". But knowing "about" is rarely useful on its own. Knowledge is of little value if one cannot use it to do something. In everyday life we need to be able to do the right things and do them well. Being able to choose the right things to do in a given situation is, of course, a "knowing how" skill, not a "knowing about" skill. Even reflecting on one's own choices and expressing reasons why one would choose a particular path require "knowing how" skills (in this case knowing how to think about one's own actions and knowing how to express oneself to others), not primarily "knowing about" skills.

The researcher's own "knowing how" skills developed over a period of several years working alongside more experienced developers, analysts and managers, learning through observation. Having to work with customers challenged assumptions. It taught the importance of effective and consistent communication in helping to design systems that suited *people.* The result was a habit of intuitive and automatic application of technical and inter-personal skills. A characteristic of "knowing how" skills is that you may be quite unaware that you have them, or that they are significant. Once you can do something, it seems simple. Driving a car is a difficult skill to learn but is easy once you have learned it. Running a JAD session or helping a group of business people to design a new corporate intranet may be daunting to the novice but is straightforward once you have learned how. Object modelling is baffling to the uninitiated but obvious to the expert.

It was only after teaching and research began that a conscious awareness of these skills developed. Presumably, an unconscious process was set in train in which the intuitive skills and knowledge were organised, in response to the theoretical and formalised treatment afforded to subjects like systems design in textbooks. It became obvious that one's own way of working, and the underlying beliefs and assumptions that guided it, were divergent from the mainstream curriculum in conceptual modelling and in some broader areas of information systems development. One's home-grown philosophy of information systems development was not the same as that in the textbooks. Subsequently it became clear that many others held similar views, on a range of subjects, that also differed from mainstream "received" information systems practice. Soft Systems Methodology (Checkland and Scholes 1990) and the participative methods (Mumford 1996) are evidence of this diversity of thought. But in the specific chosen field of research, conceptual modelling, there seemed to be nobody who was willing to question the status quo. At the time many researchers and practitioners were developing and proposing rival conceptual modelling methods. The subject was of general interest because of the excitement about CASE, the rise of relational database-based application systems and the growing enthusiasm for object-orientation. But academics and practitioners were all saying roughly the same thing about conceptual modelling: that it should be conducted by experts in a precise, almost rigorous way, using techniques that are almost graphical versions of mathematical notation. Likewise, the CASE tools designed to support conceptual modelling were tools designed with expert users in mind.

The point about mathematical notations is an important one. When relational theory emerged in the 1970s, it was considered a real breakthrough—finally, some genuine scientific theory had been developed to underpin a part of the hitherto woolly and ill-defined subject of information systems. To most this was self-evidently a positive development. University academics and professionals seized upon relational theory, adding related developments such as entity-relationship modelling (1976), structured analysis and design (1979) and, later, formal methods such as Z and VDM (Dawes 1991) which really pressed home the mathematical connection. Because these approaches seemed so successful, it was expected that information systems development could be reduced to the application of the new methods. The emerging CASE tools would automate the process and many confidently predicted that applications programmers would disappear as a breed before too long. At the most basic level the thinking was this: *It is not possible to design information systems without first deciding precisely what items of information are needed and then defining exactly how the items of information are related to each other. Any mistake in this process will lead to faulty information systems and will be time-consuming and expensive to rectify.* This thinking is linked to the idea of software requirements as something that exist *before* software is produced. This way of seeing requirements is prevalent, arguably because it is convenient for software developers to ask their clients to work that way. But it is a rather questionable idea. Perhaps a more realistic view would be that requirements rarely exist independently and that they usually come into being through a process of negotiation. The question "What do you want?" is answered by "What can we have?" or "What is practical/within our budget/available quickly/politically acceptable?" Whether or not requirements really exist before one embarks on systems design, it is often difficult to decide exactly what information is required, and a comprehensive analysis of information requirements of the sort discussed above is often difficult to achieve.

A key insight at this stage was to make the connection between what was happening in the discipline of information systems and what was going on in the world of psychology. At the same time that relational theory and related, mathematically based modelling techniques were emerging in the information systems field, there existed an independent but parallel trend in the field of psychology. The development of cognitive psychology was making it acceptable to question how the mind works. The emergence of cognitive psychology was linked to a popular perception of the brain *as a computer*. A computer consists of a central processing unit together with memory, disk storage and peripheral devices. By analogy, it was argued, the brain has short-term and long-term memory as well as its connections to various sensory organs. The idea of the computer as an electronic brain, or the brain as an organic computer, is still prevalent today. But this view is becoming hard to sustain; the more we discover about brains, the more we see that they work quite differently from computers.

Continuing in the "brain as computer" vein, cognitive psychologists postulated in the 1970s that the mind forms *mental concepts*. The idea of mental concepts is similar to the idea of a computer storing data. For instance, if we have a mental concept 'cat' then we can store information about particular cats such as their names, colours, breeds, owners and so on. However, to do so we must have some other mental concepts: the concept of a name generally, the concept of colour, and so on. This is of course very similar to the problem of storing data in a database, in which the database structure must encapsulate various data types and categories depending on the subject matter of the data to be stored. The idea of mental concepts was so self-evident to cognitive psychologists that it hardly seemed necessary to justify it, and it is an idea that still has currency today. Cognitive science as a field is largely based on the idea that our minds store and relate mental concepts. The obvious next stage was to question how it is that our minds can so easily jump from one concept to another; how we can be *reminded of* one thing by something else. In answer to this question, cognitive psychologists postulated that the mind forms *semantic links* between its mental concepts. Various mechanisms were proposed to explain this linking, including hierarchical classification schemes, semantic networks and conceptual schemas (see Appendix B for a full analysis).

In being introduced to the theoretical literature of psychology, it became apparent to the researcher that relational theory and allied developments such as semantic data modelling were strikingly reminiscent of psychological ideas about mental concepts and semantic networks, if not derived directly from them. The timing of Chen's landmark paper (1976) on entity-relationship modelling, for example, and the early development of object-oriented ideas both suggest that they were inspired by thinking from early cognitive psychology. It could be argued that these developments in psychology and information systems were a product of the rise of computers and the development of the "brain as computer" analogy in the 1960s and 1970s.

Specifically, the fundamental principles of today's standard conceptual modelling methods (in the narrow sense: object modelling and data modelling) were developed during that period and show a marked similarity to then popular psychological ideas about mental concepts, semantic networks, classification hierarchies and inheritance of properties. Many of these ideas are rooted in the 'brain as computer' model. But this model is today giving way to an alternative view of the mind, based on the operation of biological neural networks, just as earlier models of the mind gave way (mind as telephone system, mind as clock, etc.) In a neural network there are no mental concepts and no semantic networks. The neural network recalls information purely on the basis of similarity. Information is not stored at any particular location in the neural network; 'knowledge' is distributed throughout the network. Because of this it is difficult to search a neural network for specific facts. The network will retrieve information in response to a stimulus but, in general we have no way of knowing whether it is the 'right' information or not. The network is incapable of 100% correct recall. But the network can continue to function effectively even after large parts of it have been destroyed. If we accept the neural network view of brain function, then the 'brain as computer' model breaks down. Mental processing cannot occur in a logical, sequential manner. By implication, theories about semantic networks and mental concepts also become much less plausible. And while this does not negate relational theory and related conceptual modelling techniques as useful system design techniques, it suggests that they do not actually mirror very well the way people really think. It renders them a less obvious choice for modellers—who need to involve and communicate with business end users in a way that they can readily identify with and understand.

This thinking gave concrete form to the researcher's misgivings, mentioned earlier, about the received wisdom on information systems design (McGinnes 1992). The frustration with contemporary system design methods and their seemingly outdated basis developed into the general idea behind this research: that various aspects of contemporary psychology, particularly the connectionist model of cognition and recent work on mental models, could inform today's system design process (McGinnes 1994c). The essential idea was to stop thinking about modellers and business users who participate in modelling as rational information processors (the "brain as computer" analogy) and to start thinking about them as intuitive, social beings who understand the world through interpretative perceptual processes (the "brain as neural network" analogy). Changing the perspective in this way seemed to have the potential to open the way to new and potentially better design methods. At this point, and with this motivation, the doctoral research proper began with a thorough literature survey.

Theory generation and development - Literature survey

A literature survey was carried out in each of the two main relevant disciplines: (a) information systems development methods (together with their associated conceptual modelling techniques), and (b) cognitive and group psychology. In information systems, several different classes of information systems development method were critically examined. Analysis used a framework that categorised methods into thirteen groups and focused on several relevant factors: (i) their overall approach to model construction; (ii) implicit assumptions; (iii) the extent of the system development task addressed; (iv) specific steps that apply to conceptual modelling; (v) the context and representation of conceptual models produced using the methods, if any. Results from the literature survey are presented in Appendix A. The thirteen groups included relatively recent developments such as object-oriented methods (Rumbaugh 1991) and DSDM (Stapleton 1997), more traditional approaches such as SSADM (Bryant 1995), and less well-established but noteworthy methods such as Soft Systems Methodology (Checkland and Scholes 1990). The result is a reference source on past and present information systems methods as they apply to conceptual modelling.

In psychology the literature review consisted of a rather broad sweep of cognitive and group psychology, resulting in a summarised account of relevant theories and findings. The scope of the study was dictated by what was potentially relevant to conceptual modelling. The aim was to allow for the widest possible range of contributions that psychological ideas could make to conceptual modelling, so that specific applications could be selected at a later stage. In practice this meant that work was included if it conceivably had a potential impact on either the representation or content of conceptual models, or if it could offer insight into the process of model development itself or the ways in which conceptual models are used. Results are given in Appendix B. The review of the psychological literature used a framework that classified work into one of four main areas: (i) memory, (ii) perception and comprehension, (iii) mental models and (iv) group psychology. The first three categories fall under the general category of cognitive psychology. The fourth category concerns group dynamics and the properties of groups versus those of individuals.

In the review of the literature, very recent theories and research findings that had yet to become widely accepted were not used. A main aim of this work is to see whether more recent ideas can be introduced into conceptual modelling techniques. It would not make much sense, therefore, to introduce fringe theories or premature results. In reviewing the literature we concentrated on reasonably well established, mainstream thinking with widespread acceptance over the last five to ten years. For example, the neural network view of brain function has developed over a reasonably long period (primarily in the last two decades), there is a good body of evidence to support it, it is widely accepted, and no rival theories have yet emerged to challenge it. We have covered it in Appendix B and go on to use its results to guide the formulation of the modelling principles. But earlier theories about the existence of fixed mental concept hierarchies are now arguably giving way to newer ideas. Although some of these theories are presented in Appendix B, for completeness, we do not use them in formulating the modelling principles.

This part of the research was crucial in that it was the first stage at which the thinking behind the research could go beyond anecdotal and personal experience. To introduce some rigour into the process and to help ensure a consistent and thorough approach, suitable frameworks were used as described above to analyse the respective literatures. As might be expected, the frameworks evolved during the process as new information came to light. It was important at this stage of the research to avoid being overly selective, since a biased literature review would have had a major impact on the subsequent stages. It is all too easy to find what one is looking for without genuinely considering alternative views and weighing the objective evidence. Some degree of selectivity was necessary and unavoidable given the potentially enormous scope of the two relevant literatures. But staying close to the agenda established by the respective frameworks, systematically following up relevant references, and paying explicit attention to a careful and methodical approach, all helped to ensure that a comprehensive and thorough literature survey was conducted.

The literature survey brought to light many ideas that were not part of the original thinking behind the project and, inevitably, these ideas affected the underlying conception of the research. Although the results of the literature survey have been reported as objectively as possible in Appendices A and B, and are used explicitly in subsequent stages of the work, they also had a major impact on the fundamental ideas that have shaped the research at a very basic level. For example, the work on group psychology revealed many interesting properties of individuals and groups that offered the potential for exploitation by suitably designed modelling techniques. The work on mental models prompted new thinking around the central idea that conceptual models can be thought of as real-world embodiments of concrete, analogical mental models. The research on memory and perception explained and provided a theoretical basis for many ideas and observed properties of modelling and modellers that had previously been understood only from an intuitive, practical perspective.

Theory generation and development - Conceptual study

The results of the literature survey were used to derive a series of approximately thirty modelling principles, in the form of heuristics or rules of thumb. Each principle is a concrete suggestion about conceptual modelling practice. The principles are divided into three groups, covering model content, model representation and the modelling process. Every principle is supported by a reasoned argument that calls upon relevant evidence from the literature survey. The principles do not in themselves define any particular way of modelling. Instead, they are stated at a level that makes them capable of general application to different conceptual modelling techniques. The principles, which are stated in Chapter 3, could be thought of as suggested "proverbs" for conceptual modellers.

Together, the principles form a body of theory to be tested in the subsequent stages of this work. The aim in formulating the modelling principles is to create a succinct set of guidelines that can manageably be applied in modelling experiments. Each principle thus summarises a large body of evidence. Although arguments are given in support of the principles, one cannot really consider either the arguments or the principles themselves to be true or false in any absolute sense. One may either accept each relevant argument and the conclusion drawn from it, or disagree. Obviously this work is built upon the principles and therefore they are not offered lightly. But the argument is necessarily subjective and has been presented as fully as possible so that the reader may form his or her own opinions. The modelling principles are used to help understand a simple conceptual modelling technique (method 'X'), which was used later in the experimental part of the research. This stage of the work is like a thought experiment that asks: *"What could the practice of conceptual modelling be, if we dispensed with current practice and started again from first principles?"* The modelling technique and an explanation of how it works are set out in detail in Chapter 3. A software tool was used to support the modelling technique.

This stage of the research—formulation of principles and method—is probably the most contentious from a methodological perspective. It involves an essentially creative and speculative type of research that can make *"a valuable contribution to the building of theories which can subsequently be tested"* (Galliers 1992). But it also risks becoming a rather "unscientific" process. For example, the modelling technique arguably complies with the modelling principles, but it would be impossible formally to infer the technique from the principles using a logical process of deduction. The modelling technique is best seen as only one possible interpretation or embodiment out of many possible interpretations or embodiments of the principles. In the same way, the principles themselves are derived from the psychological evidence through subjective (though reasoned and explicit) argument. The idea is to offer a clear and plausible justification for each particular interpretation of the psychological evidence; in other words, to be as objective as possible. We place a particular, reasoned interpretation on the evidence, considering and weighing up where necessary the alternative interpretations that could be made. The chosen interpretation is then used to justify selected innovations in modelling technique.

One main way of introducing a degree of rigour into this type of research is by maintaining a high degree of openness and transparency (Silverman 1993, Spradley 1979). Although the argument is inevitably subjective, it must at all times be laid out plainly, with reasoning explicitly stated. No conclusions may be drawn without stating the justification and no deductive leaps may be made without giving explanation. Unsupported claims are to be avoided. We have taken this approach by applying systematically the same procedure to a series of subject areas in turn: the evidence is first summarised and then an argument, based on the evidence, is developed that culminates in the expression of a modelling principle. Each modelling principle is intended to encapsulate the main significance of the evidence on which it is based. To do this requires a comprehensive and even-handed treatment of the literature in which all options are considered and the evidence is weighed before each principle is formulated. The results are then presented using a simple framework, avoiding what could otherwise become a rather jumbled and arbitrary organisation of ideas. The reader, therefore, has the chance to understand the argument and to form some view of it.

Theory testing - Field experiment

Because the experimental design is discussed at length in Chapter 4, here we shall consider it only briefly. A detailed critique of the experimental method is given in Chapter 6. A field experiment was conducted in which method 'X' was applied, alongside a conventional conceptual modelling technique, on a small number (nineteen) of genuine business problem situations in business organisations. Models were produced by expert and less experienced modellers, including complete novices, working both in groups and individually. Modellers were selected according to their level of business knowledge. To help ensure a reasonably equal comparison between method 'X' and the control method, parallel groups of modellers were used. The modellers in each group had similar levels of relevant work experience and training.

This stage of the work is perhaps best characterised as a large field experiment consisting of a series of smaller field experiments. We have already discussed the rationale for experimenting in the field rather than the laboratory. There is no doubt that the researcher can exercise greater control in laboratory experiments than in field experiments. However, the realism attainable in a field experiment can be a great advantage. In our case we were interested in conceptual modelling taken as a whole and in its in proper context, rather than some limited aspect of conceptual modelling. It would therefore be rather pointless to attempt a laboratory experiment, since it would have to comprehensively replicate business conditions, making it essentially identical to the equivalent field experiment. The business context would need to be preserved and the participants would still need to be selected on the basis of prior knowledge of the business areas to be modelled.

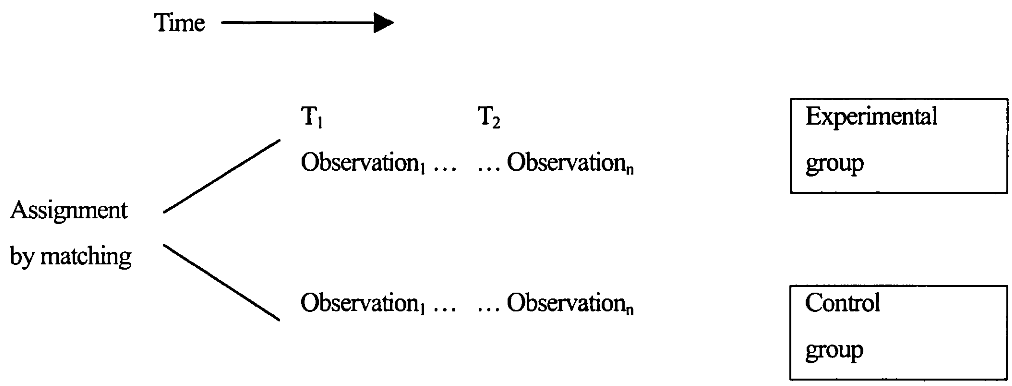


Figure 2.2 Structure of experiment (after Bryman 1989)

The chosen experimental design uses a common structure, which can be described as *equivalent control group experiment with pre- and post-testing* (Bryman 1989). In our case the pre-testing consisted of completion of questionnaires by the modellers involved in the experiment. The post-testing included participant observation, further questionnaires, interviews and analysis of the resulting models. In the classical experimental model, subjects are chosen for participation in an experiment at random from a large homogenous population. However, in organisational studies it is frequently difficult to avoid non-random selection. For example, it may be that the participants are chosen by management rather than by the researchers. Only certain staff may be available or willing to participate. The non-random groups selected in these experiments are sometimes described as "naturally occurring". A common substitute for random selection is matching subjects in terms of known characteristics such as work experience, gender, type of work undertaken, and so on. This is termed a *quasi experiment* (Cook and Campbell 1976). As Bryman observes, the quasi experiment is *"undoubtedly an inferior method of establishing equivalence, since you can only match in terms of factors that occur to you or are evident"*. In some cases the use of matched subjects is considered sufficient to invalidate the experiment (Kidder and Judd 1986). However, there are also certain advantages to matching over random assignment. In the case of small populations from which subjects are to be chosen, a random assignment may well produce between groups a seriously unequal distribution of knowledge, skills, background, and so on. It may require careful thought and adjustment by the researcher to create well-balanced and equivalent groups.

Indeed, it can be argued that the initial screening of potential candidates prior to random selection effectively constitutes a form of matching (Silverman 1993). Matching is certainly a commonly used approach in organisational field research and therefore we chose to use it, albeit with caution. In our case the modellers in the nineteen modelling studies were drawn from different organisations and, as a group, exhibited significant variation in age, training, gender, prior IT experience and business knowledge. None of these individuals could be described as the "average" office worker. It was obvious that a random assignment of the modellers to groups could easily result in serious bias and so under the circumstances, matching was chosen as the safest approach. The various potential sources of bias and the steps taken to avoid them are discussed in detail in Chapter 6.

During the experiment, results were gathered by various means. Part of the field experiment involved observation of the behaviour of modelling group members and subsequent interpretation of this behaviour in the light of other evidence, as outlined in the next section (Urquhart 1997, Walsham 1993). Interviews and questionnaires were used to gain deeper insight into specific issues. We also analysed the results of modelling itself (Elan, Walz, Curtis and Krasner 1991) in an attempt to uncover developmental patterns and other hidden information. Because this part of the research combined several methods, specific precautions were taken to ensure that best practice was observed for each method. These measures are summarised below.

Interview The modeller who conducted group modelling sessions was interviewed using a semi-structured interviewing style. The aim was to find out what the modeller felt was important, especially with regard to the conduct of modelling sessions, the tools and techniques that were used, and the outcome of modelling. Full interview notes were kept (Appendix C). A summarised and annotated version was prepared highlighting the main points, and this is contained in Chapter 6. In the notes and the summary the interviewee's comments were treated not as objective fact but as situated accounts that were likely to be affected by a range of factors other than modelling itself. Accordingly, care has been taken to distinguish the interviewee's own statements from those of the investigator, as suggested by Silverman (1993). Interpretation of the interview data is discussed in the next section.

Questionnaires were formulated with care so that their questions would be meaningful to the respondents. In this study we were fortunate that the respondents were a small, closed group whose general background and current working conditions were known. Therefore it was possible to anticipate with some confidence the way in which questions would be interpreted. When the questionnaire responses were coded and analysed, care was taken to avoid treating respondents' comments as literal statements of fact. Instead, they were treated as situated accounts that were potentially influenced by context and reflected the (a) the respondents' own perceptions and unique perspectives on the situation; (b) whatever image the respondents hoped (subconsciously or consciously) to put forward about themselves, and (c) the perceived expectations of the researcher, their employer and their peers. One problem with pre-testing in this way is that it can sensitise subjects and so create a greater effect than would otherwise exist. It is possible to overcome this problem by including control groups that are not pre-tested. However, in our case the pre-testing was judged to be unlikely to create a significant sensitising effect since it focused only on the personal details and past history of the potential participants. Conceptual modelling abilities or knowledge, which are the subject of the experiment, were not pre-tested.

Numerical analysis When it was necessary to analyse the contents of models, much thought had to be put into ways of evaluating models and tracking their development that were as objective as possible. Each model was produced for a different application domain (business area) and hence it was almost impossible to lay down in advance a "correct" form for the models. Instead the criteria given in Table 2.2 were used to assess the correctness and completeness of models. The act of applying the criteria was itself open to various influences including the prejudices of the evaluator. However, conscious attention was paid to the need for an even-handed and fair analysis in an attempt to avoid significant bias. Chapter 6 discusses this issue in detail.

The overall method of assessing models was a two-stage process. First, the final version of each model was considered and any corrections agreed with the author of the model. Secondly, each version of each model was compared exhaustively against the corrected version. This created a series of snapshots, each evaluating a model at a particular stage of development. Over one hundred model versions were analysed in this way. Although it was extremely time-consuming, one advantage of this method of assessing models was that it was difficult for the evaluator to see "the big picture" while evaluating each version. Any sense of the outcome, except a broad and not necessarily accurate appreciation of model quality, was impossible. It was only when the results were plotted in graph form that patterns of model development, and correlations between models were observed, in many cases confounding "common sense" expectations.

In addition, a limited secondary experiment was carried out in which a larger group of inexperienced modellers produced models similar to those in the main study. The resulting models were analysed to provide data for comparison against the results of the primary study. The secondary study used only a subset of the measures that were used in the main experiment. These measures focused on model completeness, a rough overall indicator of model quality. One risk with this form of triangulation is that the measures being compared may not be identical (Silverman 1993). However, in our case care was taken that exactly the same calculations and criteria were applied in both experiments for the measures that were used. In addition, the experimental conditions were similar. The significant differences between the main and secondary experiments were noted so that they could be taken into account. In particular, one major difference was that the modellers in the secondary experiment were all computer science students, while only one of those in the main experiment was currently a student. The effect of this and other differences is discussed in Chapter 6, where a detailed critique of the experimental method including primary and secondary experiments is given.

Participant observation In research involving observation, the researcher has the choice of a number of roles (covert/open, active involvement/passive observation). We chose a combination. As already mentioned, the experienced analyst who conducted group modelling sessions was not a trained researcher but was able to observe the group participants covertly and report back his observations. The modelling sessions were taking place as part of a genuine project and the group members themselves were unaware that they were being observed. The modeller was himself observed (with his knowledge) using a video recording of selected modelling sessions. In individual (i.e. non-group) modelling sessions the experimenter participated directly, observing the behaviour of the lone modellers. The lone modellers knew they were participating in a study, but did not know the extent to which their actions or their models were subject to scrutiny.

It was unavoidable therefore that modellers were informed to some extent about the fact that they were participating in an experiment. Reactive effects such as the Hawthorne effect (Roelthisberger and Dickson 1939) often mean that the actual performance measured differs from that to be expected under normal circumstances. For example, it has long been known that experimental subjects often modify their behaviour in order to support the hypothesis around which an experiment is organised (Orne 1962). This was less of a problem in our study because we were interested in comparing the effects of two modelling techniques, rather than measuring their effects on any absolute scale. The actual level of success or failure which participants experienced was not particularly material (and we have no absolute scale on which to measure it in any case).

The possibility of reactive effects and their likely consequences are discussed fully in Chapter 6. For business users in group sessions one way in which we countered the possibility of reactive effects was by using covert observation, as mentioned above, in which the group members were not informed about the experiment they were participating in. Confidentiality was aided by the nature of the experiment, which was a series of separate field experiments involving people did not otherwise meet one another, rather than one large experiment involving a large group of people. The ethical position with regard to covert observation is discussed in Chapter 4.

A possible alternative research design would have been to use a "double-blind" experiment in which neither the modeller nor the observer know whether they were part of the control group or the experimental group. This approach would be an effective way of guarding against prejudice on the part of the experimenter, which has been shown in a number of studies to be a prime source of bias (Rosenthal and Rubin 1978). However, it would have been difficult given the nature of our experiment.

We can test the quality of this research by looking at reliability (internal and external) and validity. One important question is the extent to which results are *externally reliable* (i.e. generalisable) (Bryman 1989). For instance, could we expect similar results in similar organisations? In different types of organisation? Would other modellers show the same patterns of behaviour? The experiment was not a truly longitudinal study, and so it was difficult to test external reliability. However, the experiment did incorporate a limited test of external reliability because it consisted of a series of linked modelling sessions that spanned the course of several months. In addition, the secondary study allowed triangulation since its results could be compared with those of the primary study. Both of these points are taken up in Chapter 6. With regard to internal reliability, we can calculate the average correlation between the facets we have used to compute derived measures. For example, we combined separately determined measures of completeness, correctness, accuracy, and so on into a single factor 'effectiveness'. In Chapter 6 we calculate the internal reliability of this measure. Internal reliability was also assessed through triangulation since we compared the results of model analysis with data from participant observation and questionnaires. Validity is considered at length in the analysis in Chapter 6. In our study we have two main groups of numerical measures: those resulting from analysis of questionnaires and those derived from analysis of models. The measures from the model analysis are obtainable through direct inspection (i.e. by comparison) of models. Since they are used to measure abstract concepts (correctness, etc.) we must assess their validity. We have assumed that model correctness and completeness can be determined by comparing intermediate versions of models with their completed versions, and that each model reaches a state which we can reasonably call 'correct' after modelling is complete and following any necessary further changes by the researcher. In fact there can be considerable disagreement between conceptual modelling experts about the best way to model any given situation. Whether a given model is actually correct could therefore be open to interpretation. Two models may be syntactically different but semantically identical (i.e. expressed differently but signifying the same thing). On the other hand, models may be both syntactically and semantically different if the modellers' interpretations of the business system are different. For the purposes of our experiment the researcher endeavoured to steer the modellers towards a standard, 'vanilla' modelling approach, based on the presumption that a database-style information system was required. However, stylistic variation and necessary minor differences due to the varying concepts of the modelling techniques were respected. Correctness was judged with respect to the final, agreed version of each model rather than any arbitrary standard imposed by the researcher.

Theory testing - Interpretation of results

The interview notes, participant observations and questionnaire comments were examined (Walsham 1993) in the light of numerical analysis of models and questionnaire responses. The results were used to formulate arguments for or against possible interpretations that are presented in Chapter 6. The overall aim of the research was then discussed in the context of the experimental results. The impact (positive or otherwise) of the specific psychological principles and other factors in the experiment was assessed.

Interpretation of interview notes is presented in Chapter 6, where the data from various sources is combined and compared. The interview notes were especially valuable in that the modeller had been involved in a large proportion of the modelling sessions (all of those involving groups of business users) and therefore had been able to develop a considered view. He also had significant experience of conventional methods that was useful for comparative purposes. Because of his level of expertise, he was able to spend a good deal of time observing the business users in the modelling sessions, and in this respect he acted as a participant observer. However, he was ultimately part of the experiment and, as one of those observed, his comments were not taken at face value.

We were also careful to interpret the measures from the questionnaire analysis in a valid way. In the case of the first five measures (see Appendix D), we could reasonably expect participants to report their actual previous experiences reasonably truthfully. However, the remaining measures all refer to *assessments by the respondent* of something. This means that we cannot take them at face value. The implication is that we must interpret only tentatively what participants say, in the light of the actual performance as determined by observation and model analysis.

We attempted to minimise the reactivity problems discussed above by using questionnaires rather than interviews, and the modellers were not told about any particular expected result. However, the possibility remained of some kind of unconscious bias on the part of modellers. The main counter-measure against this risk was to choose modellers with no vested interest in the outcome of the study, other than the hope that it would be a success in conceptual modelling terms. The one modeller who used both modelling techniques was directed to treat them equally.

To evaluate overall usability both qualitative and quantitative results were considered, drawing together several strands: (a) the subjective impressions reported by modellers and participants in modelling sessions; (b) the observations by the experimenter of modelling sessions, and interpretations placed on these impressions; (c) numerical measures of modellers' productivity and accuracy; (d) measures of model quality as outlined above; (e) an assessment of the relative skill and experience levels of the modellers; (f) interview notes. The impact of other factors, such as variation in subject area complexity and skill level of group members was also considered. A full analysis of such factors is given in Chapter 6. Prior knowledge and training, in particular, were likely to influence modelling effectiveness, and for this reason we considered the effectiveness of experienced and inexperienced modellers separately. Usability was considered for four distinct sets of modellers, as shown in Table 2.2.

Experienced modellers using conventional methods	Inexperienced modellers using conventional methods
Experienced modellers using method 'X'	Inexperienced modellers using method 'X'

Table 2.2 Four sets of modellers used in study

Interpretation of the experimental results requires a certain discipline. For example, preconceptions about the applicability of psychological ideas had to be *bracketed* (i.e. placed to one side) during this process (King 1994). The aim was to understand, from the modellers' own perspectives, how they perceived and experienced the act of modelling, and to contrast this understanding with a reading of the results as manifested in the models that they produced. This required a local interpretation (Schwartzmann 1993) of meaning, derived from a rich set of data about each modeller's actions, impressions and outputs.

A number of measures were used during the experimental research and these 'raw' measures were used to calculate several high-level measures to help in interpreting the results. Three of these high-level measures were *modeller effectiveness*, *model quality* and *modelling technique usability*. By 'usability' we mean the ease with which an average modeller is able to create a conceptual model of satisfactory quality, in unit time. 'Effectiveness' refers to the ease with which a modeller is able to create a conceptual model of satisfactory quality, in unit time, with a modelling technique of average usability. Although usability and effectiveness are not very well defined measures, some estimates of the relative usability of the modelling techniques and the relative effectiveness of modellers are useful. Specific and measurable definitions for all measures are given in Chapter 4. The question of whether a model's quality is satisfactory or not is potentially a thorny one but, in practice, a working definition can be used, as shown in Table 2.3. When assessing models we considered a model's quality to be satisfactory if the criteria in Table 2.3 were satisfied. The conditions are not absolutely measurable, but they are capable of being estimated and provide a useful guide to the subjective quality of models. Details are given in Chapter 4. In practice we found that few models, except those produced by very experienced expert modellers, were completely satisfactory according to these criteria.

Criteria	**Condition is fully achieved if ...**
Consensus	The participants in the modelling sessions (in particular, the modeller) deem the model to be satisfactory.
Completeness	The model's scope is consistent with that agreed before modelling commenced.
Correctness	The model means what the modeller thinks it means (when interpreted according to the notation being used)

Table 2.3 Criteria for satisfactory model quality

As several writers have pointed out, it can be misleading to aggregate research results obtained at a low level into a high level (e.g. average) result. For example, when looking at organisational behaviour, if variability within an organisation exceeds that between organisations, then average figures for organisations do not convey much useful information. In the case of our study, we aim to draw conclusions about a modelling technique based on the performance of a number of modellers and groups of modellers using the technique. If the performance of modellers varied widely then comparing average figures for each modelling technique may not be very useful. Therefore it was important to examine individual variation in performance quite closely, and for this purpose the use of participant observation and questionnaire techniques together with the post-experiment interviews to help tease out individual differences.

2.6 Summary

This chapter has presented a research approach designed to allow psychological principles of conceptual modelling to be expressed and tested. The research design is appropriate since it permits observation and measurement while at the same time acknowledging the essentially creative and subjective nature of conceptual modelling. The research design aims for maximum realism by setting the experimentation in the field. Working in real organisations allows for a rich and multi-faceted set of data and observations to be gathered. The research method is not designed to produce results amenable to statistical significance testing; nor can it produce measurable margins of error. Instead our goal is a rich understanding of how people behave and think when they work with conceptual models, supported by some reasonably objective measurement of the results they produce. This qualitative research approach is distinct from the classical scientific experimental method but can be equally rigorous if executed well and with care. As shown in Table 2.4 the first half of phase 1 (theory generation and development) is documented in Chapter 1 and Appendices A and B. The remainder of phase 1 and the whole of the second phase (theory testing) are addressed in subsequent chapters.

Phase	**Stage**	**Aim/result**	**Refer to**
Theory generation and development	Reflective practice by researcher/practitioner.	Initial ideas; rich understanding of circumstances	Chapter 1
	Research method	Suitable research method	Chapter 2
	Literature survey (I.S. methods)	Relevant background theories and results	Appendix A
	Literature survey (psychology)	Relevant background theories and results	Appendix B
	Conceptual study	Set of modelling principles; Justification for method 'X'	Chapter 3
Theory testing	Field experiment	Experimental design	Chapter 4
		Participant observations; Interview notes; Questionnaire responses; Analysis of models; Numerical analysis of results	Chapter 5
	Interpretation of results	Interpretation	Chapter 6
		Conclusions	Chapter 7

Table 2.4 Summary of research method

3
Theoretical Framework

This chapter uses the psychological knowledge expressed in Appendix B to develop a series of heuristic principles that can be applied to conceptual modelling. The principles apply generally to conceptual modelling techniques in the wider sense (see Chapter 1) and have relevance to other forms of graphical communication. The principles are used to design a prototype modelling technique (method 'X') which is used in the experimental part of this research (Chapters 4 and 5).

3.1 Introduction

Cognitive processes in the perception of conceptual models

The purpose of a conceptual model is to represent and communicate a shared understanding of a business area or process (Loucopoulos and Zicari 1992). A model can have no meaning if we do not understand the significance (content) of its symbols (representation). Hence it is crucial for us to understand how models are perceived and understood. Psychology can tell us much about the choice of language and symbols used to construct models, and the effects they have on mental processes (Malim 1994). The group interaction that occurs as models are developed and used is also relevant. The strengths and weaknesses of groups in problem-solving and decision-making activities are quite distinct from those of individuals (Zander 1989), and modelling techniques can take this into account.

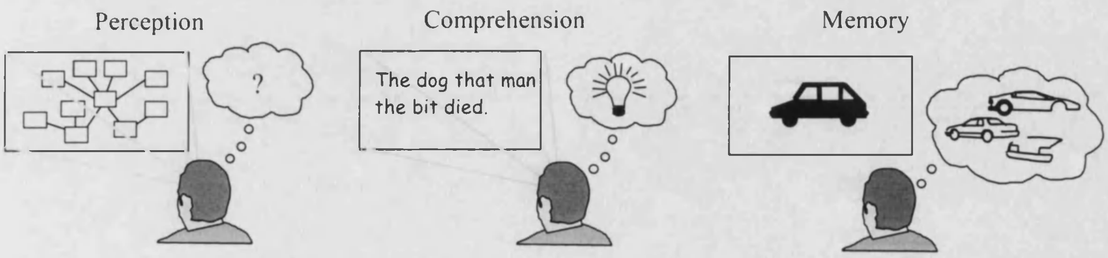


Figure 3.1 Some psychological processes used in interpretation of conceptual models

A conceptual modelling technique can be thought of as a kind of language in which thoughts and ideas may be expressed. Like any language the technique has a grammar and uses a restricted, but often extensible, vocabulary. Conventions govern the symbols used by the language and their significance. The language is useful only if there is a shared understanding of the conventions and of the symbols and of their meaning. From a psychological perspective, the comprehension or interpretation of a conceptual model, or indeed of any language, requires that the viewer's perceptions be associated to concepts already held in memory, as illustrated in Figure 3.1. When we view a conceptual model the visual sensory input is interpreted in an unconscious perceptual process. We recognise the symbols and text in the model and place an interpretation on it (Malim 1994). Familiar words and pictures stimulate unconscious associative recall of concepts in long-term memory, providing conscious meaning (Solso 1998). But we must seek explanation of unfamiliar constructs in the model before we can begin to understand it as a whole. Depending on the way the model is represented, and our level of skill, interpreting it can be an analytical act or an automatic one.

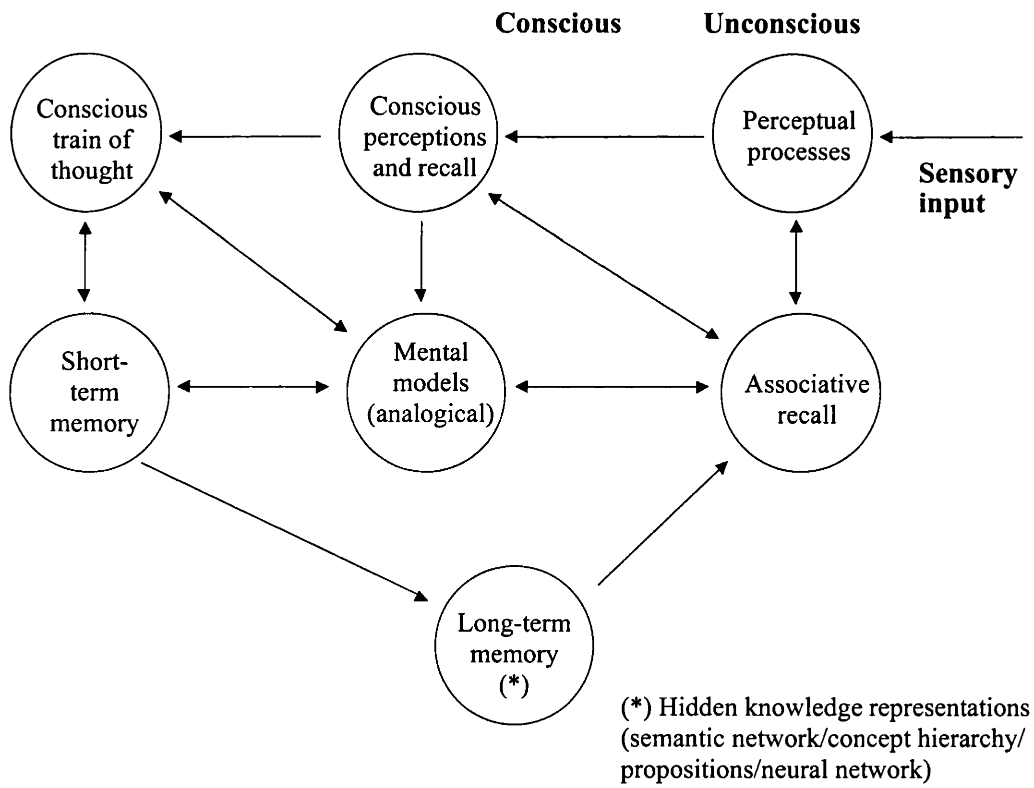


Figure 3.2 Interaction of cognitive processes during perception

3.2 Representation of conceptual models

For the purposes of the following discussion, we shall take the *representation* of a conceptual model to mean its visible appearance. To employ a computer-related analogy, the representation of a model is its 'user interface'. Through a user interface one learns about the internal state of a computer program. Through the representation of a model one learns about its structure and content. In both cases, we sometimes have to infer or assume meaning that is not explicitly represented. A representation is not necessarily identical to the model it expresses, since any model may have a number of representations, and each may be concerned with only a part of the total model. The meaning conveyed by a model's representation is subjective and so may well differ from that intended.

Understandability

Model representations embody (amongst other things) design intentions. The cost of failing to communicate these intentions can be high. Architects must prepare plans for new buildings before they are constructed. Industrial designers must create blueprints before new products can be manufactured. For similar reasons the designer of an information system must create conceptual models to describe aspects of the new system. Each type of plan or model has a dual role: to help the designer form and structure ideas about the design, and to communicate these ideas to clients and to other parties so as to facilitate discussion and ultimately to achieve agreement (Olle et al 1991). Because of this dual role, the designers of information systems are often faced with conflicting goals. To communicate a model's contents to customers, an accessible representation is required. Prose is often used for this reason—it is an extremely expressive medium that is generally understood. But some formality is also needed. Prose is a poor candidate if we concentrate instead on structuring models in the rigorous way required for detailed systems design. The flexibility of natural language makes it difficult to maintain the high level of structure generally held to be necessary in a system specification. It is easy to write indeterminate or self-contradictory specifications, which often go undetected in lengthy prose documents.

Alternatively, analysts use diagrammatic (e.g. object modelling) or mathematical techniques (e.g. the 'formal method' Z) (Duke, King, Rose, Smith 1991). Restricted vocabulary and unambiguous interpretation make these useful for system design (Loucopoulos and Zicari 1992). But the sheer 'technicality' of these techniques reduces their effectiveness in communication. For the novice analyst and untrained end user, interpretation can only be a conscious, analytical process and must therefore be slow. Many analysts and designers resort to prose translations of conceptual models when their clients fail to grasp the meaning of the models themselves (Bansler and Bødker 1993). Prototypes are widely used because they can demonstrate the consequences of requirements and design decisions. Any such concrete representation is particularly understandable, for reasons we shall discuss later. However, prototypes cannot be the sole basis by which requirements are expressed (Mathiassen, Seewaldt and Stage 1995).

To understand the tension between understandability and formality, consider a situation in which a business process must be modelled to allow introduction of supporting information technology. In an effort to improve communication an analyst might decide to model the process by recording it on video. Users can understand the 'model' easily since the video image is a direct representation of the process in operation (provided it is a visible one). But this representation is inherently unstructured and aspects which are not visually obvious cannot be shown. One cannot add internal structure to a video recording. Another analyst might represent the same process very accurately using the highly structured VDM notation (Bicarregui et al 1994). The model may capture all of the nuances and alternatives of the business reality in great detail. But a lay person would have difficulty in recognising the specification as a depiction of their own business process. The analyst would therefore have trouble in confirming whether that the specification was accurate or not.

Principle 1.1	*Choose understandable and unambiguous representations (but retain formality if necessary).*

Isomorphism

Models produced using popular modelling techniques, such as object modelling, resemble the semantic network representations used in psychology to model the apparent structure of mental concepts (Solso 1998, Green 1996). But the evidence is that they are quite different in form from both conscious (mental model) and unconscious (long-term memory) mental forms (Best 1999). Neither are conventional conceptual models *isomorphic* to (i.e. they do not resemble) the situations they describe; they are simply abstract statements of fact about situations. When an information system is designed it is essential that the designer's model of the business situation be verifiably accurate (a working definition of accuracy or correctness in this context might be the absence of misconceptions that could result in inappropriate system design). One aim of using a conceptual modelling technique is for the designer to fix in his or her mind a correct mental model of the business, so that the resulting system matches the organisation (Marakas and Elam 1998, Loucopoulos and Zicari 1992). Inaccurate mental models can be expensive and damaging to organisations.

Figure 3.3 illustrates the relationships between mental models and conceptual models during the information systems development process. The figure highlights the fact that it is by no means certain that a conceptual model will match designers' and users' own mental models. Moreover, there is no guarantee that different individuals' mental models will agree with one another. The likelihood is that they will not.

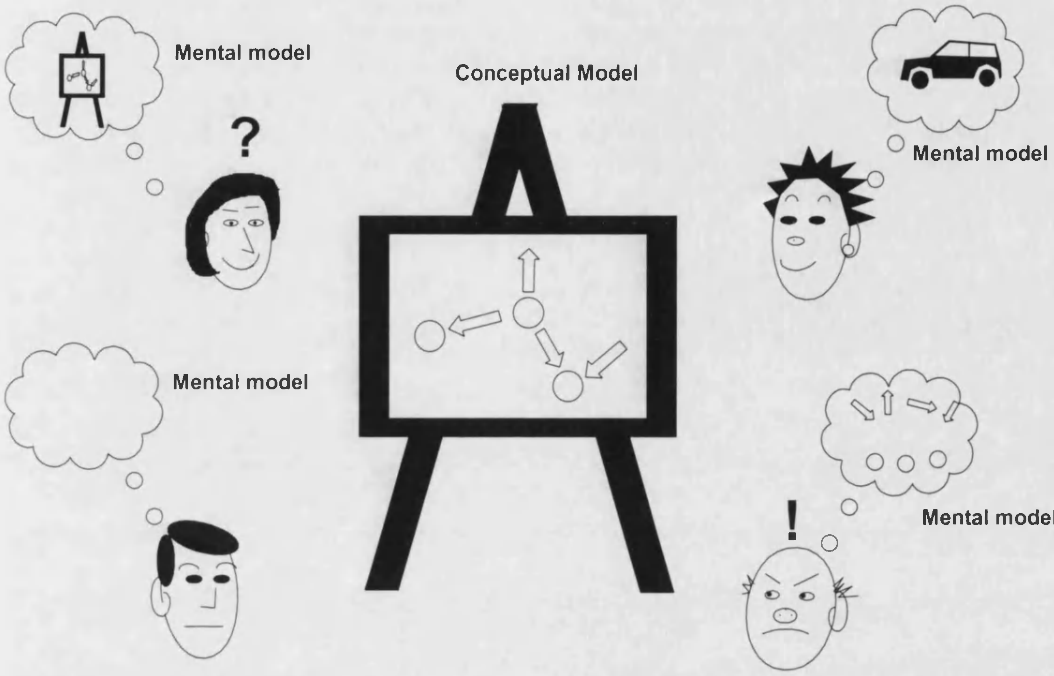


Figure 3.3 Mental models and conceptual models are not identical

Except for the smallest conceptual models we cannot take in the whole of a model simultaneously—we must focus attention successively on limited portions of it. Through 'chunking' (Miller 1975) we can remember parts of the conceptual model that we have already scrutinised, and construct a mental model of the conceptual model itself (Johnson-Laird 1993). The 'chunks' (slots in short-term memory) provide links to concepts in long-term memory. Our own mental model of a conceptual model is constructed using concepts that are either already familiar to us or have been formed as we examine the conceptual model (Solso 1998). Once we have gained some understanding of the whole model, a chunk at a time, we may be in a position to judge if the conceptual model is correct—whether or not it is a true representation of the business system it is intended to describe. To do so we must already know something of the business system and have formed a mental model of it (or of some similar situation). We can then compare our mental model of the business system with that of the conceptual model. If discrepancies are found we may wish to revise either the conceptual model, our interpretation of it, or our mental model of the business system. This comparison process is illustrated in Figure 3.4. For experienced conceptual modellers the comparison is rapid and unconscious. The modeller may well be unaware that translation and comparison are occurring, and it may not be obvious that any mental models or interpretation are involved. The skill has become 'internalised', to the point of preattentive processing (Malim 1994). In contrast, novices typically must work through each aspect of a model consciously and sequentially, much as a novice car driver has to remember each separate action to change gears in a car. Despite what experienced IT professionals may believe, significant mental effort—in conversion, recall and comparison—must be expended to interpret and verify a conceptual model. Even a correct model may be expressed in terms so unfamiliar (i.e. different from one's own mental model) that too great a mental effort is needed to convert one representation into another. Anyone who has ever tried and failed to understand a badly-written specification will agree! But if conceptual and mental models are well-matched in form then the need for mental effort is minimised. Pre-attentive processing may be possible even for novices. Much of the cognitive effort involved in searching, recognition and interpretation may become unconscious, automatic and effectively instantaneous (Marr 1982).

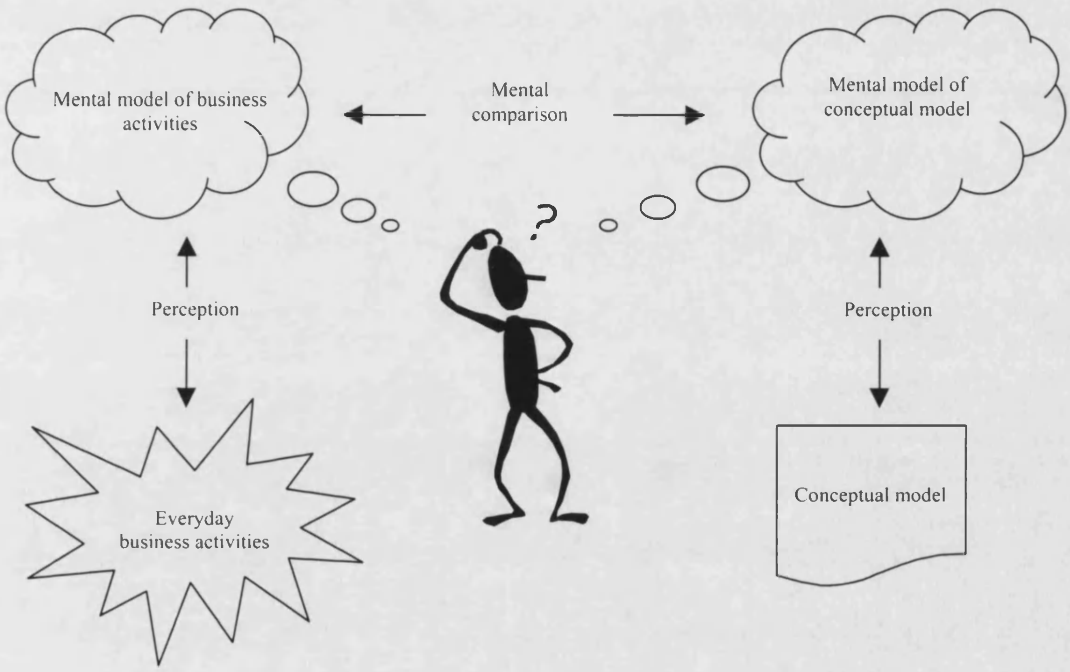


Figure 3.4 Relationship of conceptual and mental models to business activities

Principle 1.2	*Use conceptual models that match mental models.*

Terminology

To be understandable by a user, it is crucial for conceptual models to be expressed in vocabulary from the user's domain. Unfamiliar language forces the audience to devote mental effort to understanding and learning new terms, before they can understand statements that use them. The need for cognitive effort is increased and short-term memory, whose capacity is known to be very limited (Malim 1994), is overloaded. Invention or appropriation of terms is often hard to avoid in conventional conceptual modelling. The need to express concepts in a highly structured way can lead to use of terms that do not correspond to recognisable real-world concepts as far is the user is concerned. But to reduce the load on short-term memory the analyst must avoid constructing a new language simply to describe requirements or designs (De Marco and Lister 1987). The need to construct models from the user's own vocabulary should not become an excuse for lack of structure or clarity, however. Convoluted phrasing and long sentences place additional demands on memory. Neither of the two major representations for conceptual models, prose and diagrams, are immune from being unstructured, complex, lengthy, indeterminate and internally inconsistent. Material presented in a structured form is more easily assimilated, especially when organised hierarchically (Solso 1998). Chomsky postulated that sentences become harder to understand the more removed they are from what he termed their 'deep structure' (Chomsky 1968) which is effectively equivalent to the simplest possible phrasing. Hence verbal representations of conceptual models should be stated as simply as possible.

Principle 1.3	*Communicate effectively using the users' own language—in a structured, organised way.*

Using preattentive processing

With a graphical representation one can take advantage of pre-attentive processing (Marr 1982) and consequently this can be a most effective way of conveying information. Graphs, bar charts and other visual aids are frequently used to put across information. Figure 3.5 illustrates how the brain is able to gather information far more easily when it is presented in visual form. The potential for graphical conceptual modelling lies in unlocking similar capabilities in the interpretation of models. For preattentive processing to come into play, graphics should adhere to certain rules. Obvious visual features such as colour and shape may be varied to convey information and imply connections. But features should not be varied in combination since this normally requires attentive (conscious, analytical) processing.

Textual representation:

	Aug-86	Sep-86	Oct-86	Nov-86	Dec-86	Jan-87
Albany, NY	$28,675	$28,675	$29,575	$31,875	$31,675	$31,650
Memphis, TN	$28,200	$28,200	$23,400	$25,900	$22,900	$22,900
Houston, TX	$54,500	$58,000	$58,500	$63,500	$58,500	$58,500
Boise, ID	$27,250	$27,250	$27,250	$27,900	$27,250	$27,250
Minneapolis, MN (HQ)	$72,950	$74,500	$74,500	$78,500	$75,425	$77,525

Figure 3.5 Information presented in both numerical and graphical form (a)

Graphical representation:

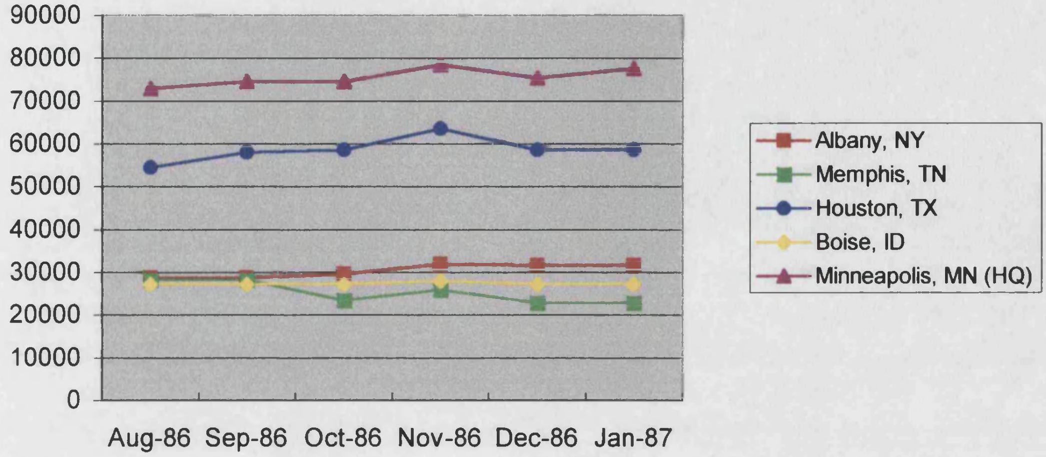


Figure 3.5 Information presented in both numerical and graphical form (b)

Principle 1.4	*Maximise bandwidth using visuals where appropriate.*

Reinforcement

Verbal and visual sensory inputs are processed largely independently of one another (Medin and Ross 1997) but the two systems are thought to interact so that recognition of images is improved when verbal cues are present and vice versa. Research into the use of imagery confirms that, for recall, combining images with words is far superior to the use of words alone (Avery and Baker 1990); an image acts as a powerful stimulus for association. The images used for this purpose need not be photographic in nature; the mind fills in details automatically, and simple but recognisable drawings can easily stimulate recognition (McCloskey and Egeth 1983, Eysenck and Keane 1995, Best 1999). Research into the processing of visual images such as cartoons has shown that the salient information extracted from a picture and used for recognition represents only a small portion of the total (Barlow at al 1990). As in graphic and graphical user interface design, visuals should be rich enough only to convey the required information.

Principle 1.5	*Maximise comprehension of model concepts by combining words with simplified pictures.*

Layout

Current modelling techniques make little use of layout and placement of symbols to convey information. Relationships between model elements are instead either explicit (as in associations between object classes in a class diagram) or implied by textual labels or descriptions (as in the connection between a data flow diagram and its 'parent' process in a higher-level diagram). Position and proximity often have no meaning in these diagrams. Where layout is important, it tends to be used in a redundant way; for example, external entities (sinks/sources) are by convention placed around the outside of data flow diagrams, although this conveys no additional information.

Gestalt psychologists demonstrated that proximity and similarity of symbols conveys meaning automatically (Malim 1994). There are several ways in which this fact can be used to increase the understandability of conceptual models, without increasing cognitive load. For example, one could depict similar concepts using similar symbols. Two concepts that together constitute a third concept could be placed in close proximity to one another, perhaps enclosed by a boundary. Size, colour and texture could also be used to convey information. Layout is also important in aiding recall. If consistent positioning is employed, we can remember items according to their locations—the 'spatial arrangement mnemonic' (Bellezza 1983).

Another automatic perceptual process is the use of visual cues to judge depth. We can interpret three-dimensional scenes with ease, understanding intuitively the spatial relationships between objects in a scene. Computer user interfaces such as VRML-based web sites capitalise on this ability. The user perceives a two-dimensional image as a three-dimensional scene. More information can be packed into the same space without overloading the viewer (Herndon, van Dam and Gleicher 1994). It is a small step to imagine three-dimensional conceptual models that users could navigate at will. Recognisable objects could be placed in a scene at some apparent distance from the observer. To manipulate part of a model the user would 'move to' the concept of interest to get a 'close up' view. Anyone who has attempted to draw non-trivial conceptual models will know that it is not easy to lay the diagrams out in a clear and coherent fashion. Avoiding crossed lines can be difficult. One useful consequence of three-dimensional conceptual models would be the prospect of banishing crossed lines as a problem for the modeller.

Principle 1.6	*Use layout effectively and consistently to increase understanding and improve recall.*

Context and other cues

Research on perception tells us that context is all-important in the recognition and interpretation of situations, symbols and language. The effect of context is subliminal and, although it has a strong influence on our perceptions, we are unlikely to be consciously aware of its effects. Context induces a *set* in the observer, a predisposition towards one particular interpretation (Avery and Baker 1990). Reinforcement between graphical and verbal symbols is one example of the effects of context. Modelling techniques can use context to make life easier for inexperienced modellers. While attention may be devoted to one particular element within a model, the surrounding graphical or textual elements have an important role to play. Modelling techniques can use recognisable graphical elements rather than arbitrary shapes (Day and Bellezza 1983). With suitable notation, conceptual models can be laid out so that related elements surround the current focus of attention, providing context to guide the modeller's thinking. The modeller need not start with a blank sheet of paper. Previously-completed models from related subject areas or reusable model fragments provide context that stimulates the modeller to think along the right lines (Bechtel and Abrahamsen 1991).

Principle 1.7	*Assist recall and comprehension using context and other cues.*

Level of generality

If they are to resemble mental models, conceptual models should be depicted in a lifelike manner (Johnson-Laird 1985). The psychological evidence is that people prefer to deal with concepts at an everyday level of generality. Photographic images can be too specific, however, especially when used to represent general concepts. For example, in Figure 3.6 one could wrongly interpret the leftmost representation as meaning a specific person, or a specific type of person, rather than the more general idea of *any* person.

Figure 3.6 Alternative ways of depicting the concept *person*

For any symbol to have meaning to the viewer it must conjure up associated ideas from long-term memory (Quinlan 1991). Therefore we ought to choose symbols that are likely to produce useful associations. Each symbol should be carefully chosen because it represents a 'chunk' of knowledge. This can be a problem for abstract concepts that have no obvious visual representation. However, for more concrete concepts several options are available, as illustrated in Figure 3.6. Best of all, end users and modellers could be allowed to choose their own symbols to signify concepts. When viewing a model for the first time, unfamiliar concepts in it will fail to produce any specific associations. But if chosen well, the symbols can produce less specific but still useful associations, permitting interpretation of a general nature. It is known that the process of recall tends to produce relevant information of a general nature when specific information is not available (McCloskey and Egeth 1983). If we do not know about something in detail, we can still make reasonable assumptions about it based on our prior experience. People without detailed knowledge of the business system depicted in a model may not be able to glean as much from it as experts, but they should still be able to read it in general terms. Choice of symbol is therefore a question of balancing specificity—to achieve as accurately as possible the desired associations—with generality, to allow recognition by non-expert viewers. Current modelling techniques do not permit this. It is very hard to make sense of a conceptual model in an unfamiliar area, because the models offer no cues of a general nature, particularly if very specific terminology has been used. They are based on a narrow range of graphical constructs, such as rectangles, circles and arrows, that lack significance for the untrained viewer.

Principle 1.8	*Use a rich set of recognisable 'lifelike' symbols—neither too general nor too specific.*

Consistency

Meaning is created when an observer perceives the individual components of a conceptual model and is then able to recall existing knowledge by association. Learning is possible when there is consistency in the way that symbols are used to denote concepts so that specific associations can be made in response to specific symbols. However, most existing conceptual modelling methods do not enforce consistent relationships between symbols and concepts. The same business concepts may be represented using different symbols at different times. And symbols are typically shared between concepts. Figure 3.7 illustrates several ways in which aspects of the business concept *purchase* might be represented. Note the multiple meanings available for some symbols.

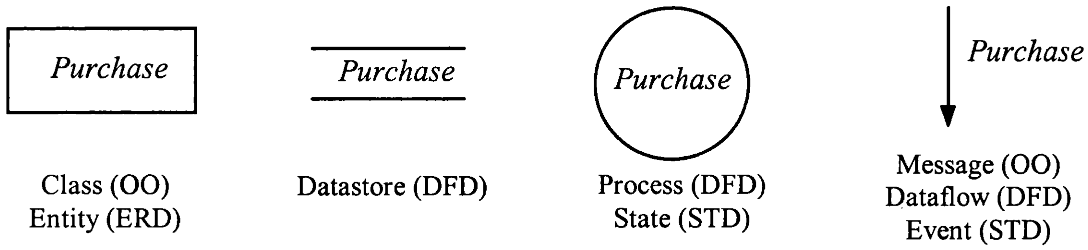


Figure 3.7 Conventional ways of representing aspects of the concept *purchase*

The different symbols represent different aspects of a purchase. The mental concept purchase itself does not appear explicitly (the closest equivalent might be the DFD process, if the DFD is a physical DFD, since a purchase is 'in reality' a form of activity). The use of different symbols means that models cannot visually be navigated using preattentive processing. The models cannot be chunked by subject; mental association between the different symbols depicting aspects of a purchase is unlikely because of this lack of consistency; one is unlikely to connect the different aspects into a unitary whole. Because pre-attentive processing cannot be used, the potential for visual searching is heavily restricted (Treisman and Gelade 1980).

Principle 1.9	*Consistently use each symbol with one and only one business-related meaning.*

Conclusion

This section has presented a number of psychological principles concerned with the representation of conceptual models. In summary, we can say that representations with recognisable symbols that denote familiar business concepts in a straightforward way, and have no hidden rules of interpretation, may help business end users to understand intuitively what is being modelled. Representations with technology-related abstractions (such as object classes), hidden interpretation rules, and a lack of obvious visual cues or context, are likely to be harder for users to understand intuitively. Interpretation will still occur, but it will probably be correct less often.

3.3 Content of conceptual models

This section concentrates on the elements used to build conceptual models and the ways in which they can be combined and interrelated.

Abstract and concrete concepts

One of the main differences between mental models and conceptual models is that mental models are *concrete*—they represent specific examples of situations and objects—whereas conceptual modelling techniques describe generalised situations and objects. For example, in a state-transition diagram, each 'state' represents a *type* of state that can apply to all occurrences of a particular type of object. In contrast, a mental model might contain a specific object in a specific state. The objects in an object model generally represent classes (*types* of object), but in a mental model each component represents a particular object. Table 3.1 illustrates the distinction for several modelling techniques. A mental model can be thought of as an instantiated schema. A component of a mental model may stand in for many entities, but is nonetheless concrete and specific and represents a concrete perception, whether imagined or not. The components of mental models represent themselves; they are not represented 'as' objects, entities, processes or anything else. They are not abstract concepts.

Diagram type	**Construct**	**Represents**	**Mental model equivalent**
Entity-relationship	Entity type	All entities of the given type	An exemplar entity
Object modelling	Class	All objects of the given class	An exemplar object
Data flow diagrams	Process	All processes of the given type	A representative example of the process occurring
State-transition	Event	All occurrences of events of the given type	An instance of the event occurring on a specific occasion

Table 3.1 Meaning of model constructs

Principle 2.1	*Model concrete concepts, not abstract concepts*

Distinguishing technological and business perspectives

Conceptual modelling techniques may seem simple enough to IT specialists but tend to be rather mystifying for end users. Perhaps one reason is that the perspectives they represent are typically defined with reference to technology-related concepts rather than business ones. Each symbol in Figure 3.7 represents one or more specific technology-related perspectives of the business concept *purchase.* Concepts like class, entity, datastore, process, message, dataflow, etc., are clearly computer-related ideas and some knowledge of computers and computer programming will help to understand them. To illustrate the distinction between business concepts and technology-related concepts, two alternative representations for the concept *viewer* are presented in Figure 3.8.

(a) *(b)*

Figure 3.8 Two alternative representations for the concept *viewer*

In the absence of other information, an untrained person would be likely to interpret these two representations in different ways. Representation (a) (the 'person' icon) would probably symbolise a person who views. But representation (b) (the word 'viewer' in a box) is more ambiguous. It could refer to a person who views, to some form of viewing device, to software for viewing images, or to something entirely different. A trained object modeller might recognise it as an object class that could be implemented in a software system to store information about 'viewers', irrespective of what viewers actually are in reality. But untrained user would be unlikely to make this interpretation spontaneously. The icon in representation (a) has associations to known real-world concepts ('man', 'person', 'viewing'), which together resolve ambiguity and improve recognition. But to interpret representation (b) correctly requires IT-related knowledge (e.g. whether the symbol is to be seen as part of a DFD, ERD or object model, and what these are). This may not be apparent to the untrained observer.

Modelling technique	**Model construct**	**Stands in for technology**
Object modelling	Object Attribute Message	Software object State variable Method invocation
Entity-relationship modelling	Entity Attribute Relationship Unique identifier	Database table Column in table Foreign key Primary key
Data flow diagrams	Process Data store External entity Data flow	Program or module File or database User, computer system Input/output data
State-transition diagrams	State Event (state transition)	Data values Transaction

Table 3.2 Comparison of modelling constructs with technological concepts

The business user's mental model is constructed of familiar concepts: people, documents, work, and so on. But conceptual modelling techniques do not start from these concepts. Conventional conceptual models require users to think about their jobs, organisations and customers as if they were part of a computer system. This technical, IT-related perspective is unnatural and confusing. One must represent each concept as an entity, an attribute, an object, a process, or some other construct. These are technology-related concepts. The data entity with attributes is very much like the idea of a database table with columns. A process on a data flow diagram is very much like a computer program. The use of technological concepts rather than business-related ones is evident in many popular conceptual modelling techniques, as summarised in Table 3.2.

Principle 2.2	*Model business concepts, not technological concepts.*

Pseudo-concepts

The constructs in conceptual models often do not reflect mental concepts in a straightforward way. One reason is that established rules of systems analysis and design can force modellers to invent model constructs that map poorly to mental business concepts. This can force users to learn a new way of thinking in order to interpret conventional conceptual models. An example of this kind of situation in object modelling is when a many-to-many association must be resolved, creating a new class. The new class may be difficult to name adequately if it has no obvious business counterpart. Figure 3.9 illustrates this kind of situation.

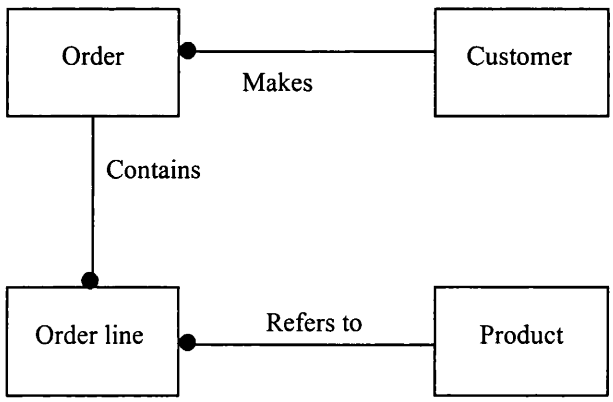


Figure 3.9 Introducing a new entity type to resolve a many-to-many relationship

In this example it is necessary to separate orders from order lines to avoid the presence of a repeating group in the order class. This standard construction is frequently used as an example in modelling texts (e.g. Barker 1990). But it is confusing for most people. The classes do not map well to 'real-world' objects. The *order* class seems real enough, although users might see an order as something that happens or is done (i.e. as some kind of activity or relationship) rather than as a *thing* in its own right. But the *order line* class is unlikely to correspond to any distinct object in the user's eyes and is more likely to be seen as part of an order itself. The *product* class could cause some confusion too. Does it refer to a particular product or to a type of product? Many users would see product type as a property of a product rather than a thing with independent existence. Structurally, the user's own mental concepts may be more like one of the two alternatives shown in Figure 3.10 (ignoring issues of representation).

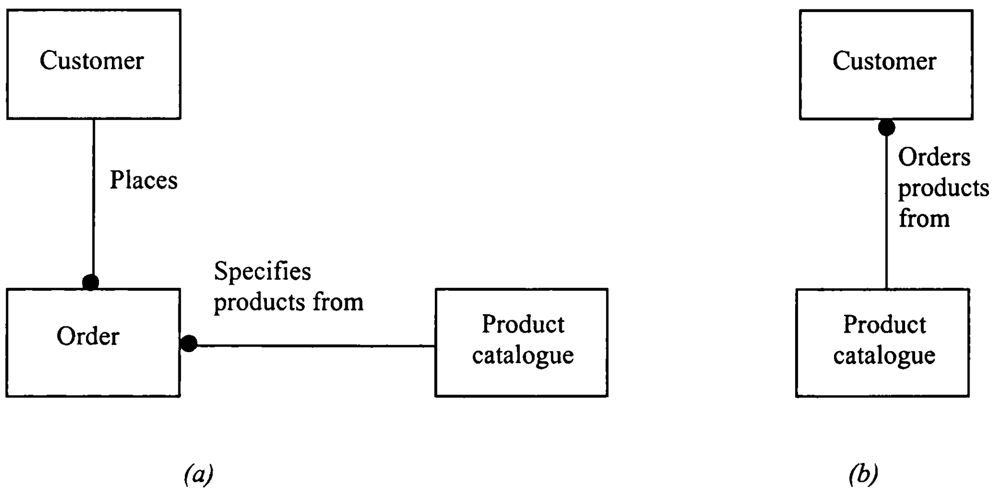


Figure 3.10 Two alternative versions

The models in Figure 3.10 misuse the notation and break modelling rules (such as normalisation) (Codd 1990). The order entity contains a repeating group of attributes (occurring once for each product ordered); in the rightmost model the presence of these attributes is hidden by the association between customer and product catalogue. Each instance of the product catalogue class actually represents a set of objects, each of which refers to a particular product in the catalogue. There may be only one product catalogue, which means that it would be unlikely under normal circumstances to qualify as a class in its own right. Trained modellers might object to the misuse of their notation shown in Figure 3.10, but it can be argued that if a model provides a recognisable depiction of the user's world, then the representation is a useful one. Strict adherence to rules like normalisation do not necessarily give us the most useful representations (Finkelstein 1990, Stamper 1987).

Some modelling techniques are not bound by these rules. For example, the LBMS client-server systems development method (LBMS 1993) incorporates a 'user object modelling' technique in which repeating groups of attributes and many-to-many relationships are acceptable. User objects are included in a model only if they are recognised as business objects by the users themselves. User objects can be thought of as views constructed from a normalised data model. User objects with no data attributes are also permitted if they are considered sufficiently important. The technique sees these 'denormalised' views as a (non-implemented) half-way house between physical data structure and user interface design.

Principle 2.3	*Introduce only concepts that exist in the user's world.*

Conceptual diversity

Users' mental models contain concepts and ideas that seem important and relevant to the users themselves. When considering a business area in anticipation of new information systems, or of renewal of existing systems, the range of important concepts and ideas can be very diverse and extends far beyond that captured by object modelling and similar techniques. As we have already seen (Appendix B), almost anything can be part of a user's mental model of a business area. An important role of conceptual modelling is to help the analyst develop the right mental model. He or she must strive to appreciate *all* of the salient facts about a situation, including more intangible factors, that can influence the success of systems development projects. The implication is that modelling techniques should allow a very diverse range of concepts to be used in creating models. But many organisations already complain of 'analysis paralysis', spending what seems like far too much time on problem definition rather than solution creation. How can they justify further effort in creating even more complex and comprehensive conceptual models? And it may be neither desirable nor even possible to highlight certain issues (such as political issues that could lead to problems in the form of open hostility or ongoing conflict).

There is some tension between the two divergent concepts of systems analysis. On the one hand, analysis should be 'wide' so as to encompass all relevant factors about the organisation, while on the other hand, analysis should be 'narrow' so as to lead smoothly and quickly into design and implementation. Few information systems methods stress the importance of capturing information other than that expressible in the form of data structures or procedures ('unstructured requirements') and many popular methods fail to address this type of information at all. It is not obvious how to represent it and what to do about it once it has been documented. But these 'unstructured' requirements can be far more important than conventional 'structured' requirements and paying attention to them may be critical (Muller, Wildman and White 1993). By ignoring them a method may force the consideration of important issues underground. Experience suggests that, in practice, the 'wide' issues are considered mainly informally and intuitively, and may be addressed implicitly in documents such as user requirements statements. The 'narrow' issues are expressed explicitly and formally in design deliverables such as conceptual models. Table 3.3 summarises the major advantages and disadvantages of including these two types of information.

	Advantages	Disadvantages
Narrow (e.g. ERD, DFD)	Easy to convert to design. Familiar for IT professionals.	Prescriptive—little opportunity to consider wider aspects. Unfamiliar for business end users.
Broad (e.g. Rich Picture)	Helps ensure that full thought is given to problems and potential solutions. Relatively understandable by users.	Unclear how to map to design. Widens scope of problem (may or may not be politically acceptable to do so).

Table 3.3 Choice of model scope

There have been efforts to create methods that allow conceptual models to surface more of the 'big' information. One example is Soft Systems Methodology (SSM) (Checkland and Scholes 1990), which incorporates the 'Rich Picture' technique. Rich pictures can include explicit place-holders to represent intangible concepts such as conflict, problems and solutions. Experience with SSM has shown that it is useful to produce models that deal with more than just data and processes. The learning that occurs during the development of rich pictures is probably helpful in forming the 'right' mental models in both users and designers. This can have a positive effect on the subsequent development of computer systems, since the designer is cognisant of more relevant information than would otherwise be the case.

Although SSM is 'rich' it is also informal in comparison with typical information systems-related methods and, like many informal methods, provides no clear transition into design. It is unclear how a systematic approach can be applied to the transformation of the information contained in rich pictures into something useful as input to modelling. Most existing conceptual modelling techniques do offer a relatively smooth and clear-cut transition to design and implementation. It is generally quite easy to state how a data flow diagram may be transformed into a program structure. The process of converting an object model into a usable system structure can be almost completely automated. But there is no simple way of converting into a design a model that may be expressed in terms of concepts like 'attitudes' and 'goals'.

Principle 2.4	*Allow models to be constructed from a rich, extensible set of concepts including unstructured information*

Personal relevance

Personally relevant information is recalled more easily than other information (Thompson et al 1987). Hence it makes sense to maximise the personal relevance of conceptual models. Choosing the right participants for modelling sessions (Zander 1989) can help (see next section on group dynamics). Another way is to ensure that the widest possible range of contributions by group members can be supported. Facilitators are trained to accept all contributions in a non-judgemental way, but modelling techniques are less forgiving, since most support only a narrow range of concepts. Avoiding the need to 'edit' contributions by allowing richer model content can help make this possible, as in brainstorming (Osborn 1957). Subjective relevance may also be strengthened if users are allowed to choose their own symbols and terms to represent familiar business concepts.

Personal relevance can be increased by allowing individuals' own model views to be captured. Conceptual modelling often aims to achieve an acceptable compromise between different views rather than representing individuals' own views. This is a consequence of the general rule that an information system is a compromise between the needs of many people. It is normally too expensive to give each person specifically what they want. Conventional conceptual modelling techniques offer no support for dealing with multiple, and possibly conflicting, views because they assume an objective world with a 'correct' view. Obviously, one is free to create as many alternative versions of models as one wishes, but the problems of integrating these models are left largely to the modeller to solve. Modelling techniques could help increase relevance by providing explicit and active assistance for capturing alternative views and for comparing and combining these views.

Principle 2.5	*Increase personal relevance by allowing users to choose their own symbols and by supporting alternative views.*

Short-term memory and attention

The limitations of attention make it easy to become overloaded by information. It is often difficult to 'see the wood for the trees'; one can focus attention on only a small number of items at any given moment. When confronted by large volumes of information we must either reduce the volume to be considered (Miller 1975) or translate it into a form that can be assimilated more easily. In conceptual modelling, information overload is a perennial hazard. Models commonly contain hundreds of separate items. The analyst must be fully aware of model content whilst attempting to make it accessible to users. We have already considered the option of translating text-based information into a graphical representation to permit pre-attentive and parallel processing. In addition, two main 'chunking' strategies are available: (a) *summarising*: making the information more general, and therefore reducing the amount of detail to be considered, and (b) *splitting*: partitioning the information into smaller sets that can be considered individually.

Current modelling techniques offer only limited support for these two strategies (see Table 3.4). Generalisation capabilities are particularly lacking since they apply only to individual concepts, not to groups of concepts. None of the modelling techniques permit generalisation about whole situations and processes, as suggested, for example, by Schank's situational and intentional memory and related work on scripts. Splitting (decomposition or partitioning) capabilities allow attention to be focused on specific portions of models—the 'divide and conquer' strategy. A large problem can be tackled by dividing it into sub-problems and then tackling each in turn. A criticism of the decomposition approach is that it does not lend itself to a holistic approach to problem-solving and can lead to poorly integrated, partial solutions. Decomposition-based chunking schemes must therefore offer integrated views to help maintain an overview.

Diagram type	**Summarising strategy**	**Splitting strategy**
Object modelling	Inheritance	Aggregation
Entity-relationship	Supertyping	None
Data flow diagrams	None	Decomposition
State-transition	Nested states	None

Table 3.4 Strategies for reducing complexity

A hierarchy is one means of chunking information that improves comprehension and retention (Bower et al 1969). It mimics an expert's organisation of knowledge and assists recall (Best 1999). Conceptual modelling techniques can use hierarchical structures for categorisation as well as decomposition, although only a small number of techniques allow both (see Table 3.4). Data flow diagrams and some data modelling techniques, such as Warnier-Orr diagrams, are based on hierarchical decomposition. Object-oriented methods include explicit notations for aggregation so that compound or nested objects can be modelled (Coad and Yourdon 1991, Rumbaugh et al 1991). Aggregation may have advantages as a grouping construct over the use of arbitrary subsets since it tends to produce more meaningful groupings of model components that reflect real-world aggregations. One is, of course, always free to partition models in an ad hoc way, simply by drawing different parts of a diagram on different pieces of paper. However, it is difficult to manage the parallel evolution (and possible reintegration) of separate and potentially conflicting sub-models. It can also be difficult to retain an overview of a model that has been partitioned. Associations between classes in different subset models become 'invisible'. Anyone with experience of designing non-trivial object-oriented systems will confirm the problems of managing a large, evolving set of mutually dependent structures (Booch 1996, Henderson-Sellers and Edwards 1994). Object modelling lacks significant partitioning capabilities and can produce large and unwieldy diagrams containing hundreds of classes. In practice, some form of informal partitioning must be used and many CASE tools that support object and data modelling also allow diagrams to be divided into arbitrary subsets. The alternative to summarising or splitting a model is to focus attention successively on different areas. The problem is then one of placing symbols or text so that related items appear close enough together to provide useful context. Doing this is often near-impossible in methods such as object modelling where every class is related to many other classes and all associations are shown explicitly.

Principle 2.6	*Support short-term memory using chunking methods (summarising and splitting)*

Fuzzy categories

Mental concepts and categories are fuzzy. We may feel that we know what we mean by a particular concept or category, but it is often difficult to say exactly what criteria we use to categorise objects (Eysenck and Keane 1995). Some objects seem to be more typical of their categories than others, and some properties seem to be more important than others in governing category membership (Solso 1998). Concepts may be defined intensionally, with explicit rules (e.g. *tricycle: a three-wheeled vehicle propelled by pedalling*). Others are defined extensionally, by enumerating the members of the group, or in terms of prototype or exemplar objects; membership depends on similarity to the prototype object (Medin and Ross 1997). Evidence suggests that we unconsciously apply a combination of methods to categorise and group objects. Context is also important: *concept instability* is the tendency of an object to be classified in different ways according to context (Eysenck and Keane 1995). Our understanding of categories may be modified by conscious reasoning, as in situations when we know that our senses deceive us ("It looks like a fish, but I know its a mammal"). But the bulk of mental categorisation is unconscious.

According to the connectionist view of brain function, categorisation occurs not through the application of rules but through the combined effects of interacting neurons. Neural networks categorise effectively—often better than humans—on the basis solely of exposure to known patterns. Like humans they learn by experience and can generalise their knowledge to new areas. They contain no 'rules'. Rule-based artificial intelligence systems are less effective at this task (Brosnan 1996). Present-day information systems are also inflexible because each item of information must be expressed in a rigid format; we cannot define database information in a vague way. Systems do not mirror the way we think because they demand explicit statement of (and adherence to) rules. Because of this it can be difficult to model fuzzy mental concepts using deterministic conceptual modelling techniques. Instead, modelling techniques could allow concepts to be defined using a combination of strategies, or remain undefined, much as we define mental concepts. They could also embrace concept instability by allowing any given model construct to be classified in a more than one way.

Principle 2.7	*Allow alternative concept definition methods and context-dependent classification (fuzzy categories).*

Level of generality

When thinking and talking about the world people tend to prefer a particular level of generality that is close to everyday experience. We are more likely to use everyday categories than abstract ones (Best 1999). Everyday categories in a business situation could include people, organisations, documents, equipment and products. More general categories are not so useful because they fail to distinguish sufficiently between everyday things. In conventional conceptual modelling techniques there is nothing special about any particular level of generalisation, however. Subtypes are defined if they have properties that differentiate them from other subtypes. If a model can be made more general without loss of information then this is normally considered acceptable. But psychological evidence suggests that this practice may make models harder to understand and verify, increasing the likelihood of errors. And conventional modelling techniques group concepts into categories like classes, entities, attributes, associations, processes and states. These are generic (i.e. high-level) concepts. A class could be described as 'something I would like to store information about'—not an everyday concept!

Principle 2.8	*Model using concepts at an 'everyday' level of generality.*

Simplicity

Users participating in modelling sessions need to understand the modelling techniques that are used. This inevitably requires learning, but cognitive load can be reduced by minimising the number of concepts and notations that users must learn. Hence it is desirable to simplify modelling techniques as much as possible. One way of doing this is by removing distinctions that have no relevance or value to the end user. For example, many modelling techniques distinguish attributes from aggregated objects, and both of these from associations. If a car has a colour, a manufacturer and an engine, then colour is an attribute, manufacturer an association and engine an aggregated object. For non-IT people these distinctions are rather fine. Even to an IT professional it may be unclear whether a given property should be modelled as an attribute or as a class. There is no evidence that we distinguish mentally between attributes (or 'properties') and classes (or 'concepts') and it is quite possible that we simply associate concepts (Best 1999). For example, we might associate the concept *my car* with the concept *Ford* and also with the concept *red*. In purely technical, object-oriented terms, every concept can be modelled as a class and there is no real need for attribution as a separate mechanism. The only classes that need attributes of their own are the base-level data types such as *number*, *text string* and *date* (Henderson-Sellers 1992). Life could therefore be simplified for conceptual modellers if the distinction between attribution and aggregation were removed. And the distinction between association and aggregation relationships is a software design decision that arguably has no place in a conceptual model in any case.

Another way of simplifying modelling is to reduce the number of distinct techniques that have to be used. Because of the way in which the techniques were developed, we now have to use one technique for modelling data, another for modelling processes, a third to model behaviour, and so on. These alternative 'perspectives' can seem to end users abstract and difficult to grasp (McGinnes 1994). Instead of alternative techniques for different perspectives, users could be presented with a single technique and notation that can represent their own 'business perspective' (i.e. reality as they see it).

Principle 2.9	*Simplify modelling by reducing the number of modelling techniques and removing artificial or fine distinctions in notation.*

Long-term memory

For requirements analysts and conceptual modellers, being able to retrieve information reliably is important. Large amounts of information are needed to create a conceptual model, and managing this information can be a challenge. A conceptual model is a complex web of interrelated facts. As a model evolves, relevant information must be located easily and new knowledge incorporated in the right places. Searching is often necessary—to identify parts of a model that might be impacted by or could be used to record new facts, or to highlight apparently contradictory or duplicated information.

Requirements statements and specifications often form the main repository for model-related information. It is easy to search through electronic documents for literal text strings using queries such as '*find every part of the document that includes the characters "sales analysis".*' Literal searching of this nature relies on objects being named according to known schemes. One must know where to look for any given item (e.g. what document to search). CASE tools generally offer literal text searching methods of this type. Searching by subject is another matter, however. More useful queries such as '*show me all the parts of the model that could be relevant to sales analysis*' cannot be answered, by word processors or by CASE. An analyst must rely on memory to locate relevant parts of a model. The analyst is able to do this because the human mind uses semantic associative retrieval. CASE tools do not. To offer similar services, CASE tools would need to operate at a semantic level. The current generation of tools deal with classes, attributes, processes, and so on, but cannot interpret the models expressed using these building blocks. Tools that could emulate the brain's use of semantic association to allow retrieval by subject would offer genuine support to the modeller who needs to navigate large amounts of information without becoming lost or bogged down in detail.

Principle 2.10	*Store models such that memory can be supported by semantic associative retrieval.*

3.4 Conceptual modelling process

The field of participatory design promotes effective group interaction in information systems development (Muller, Kuhn 1993). Client-led systems development (Stowell 1991) and Joint Applications Design (JAD) (Wood and Silver 1995) also aim to involve all participants fully in a process of interaction that will lead to improved designs. Group work is, in fact, an essential part of the design of information systems. Without consensus, no system is possible. Only the smallest single-user systems can be created without the need for discussion and agreement. The construction of a conceptual model is therefore almost always based on group work. Findings from psychology relevant to group interaction can help us to understand how conceptual modelling practice should take into account the characteristics of groups and group work. (For an interesting account of some of the problems of group modelling work, of a different kind, see Vennix 1996).

Developing and exploiting group productivity

Conceptual models are often produced by ad hoc groups constituted especially for the task on a 'one-off' basis. But groups that have become accustomed to working together over an extended period of time tend to outperform ad hoc groups (Hall and Williams 1966). Better-quality conceptual models are therefore possible if groups can work together on models over extended periods. This is normally impractical because of location or time constraints, especially given the heavy time commitment that participation in modelling often requires. Travel, lack of free time and unavailability of meeting facilities are often constraints when modelling sessions are planned.

We might envisage the use of suitable computer-mediated facilities to enable a form of 'remote modelling' in which members of a virtual group can model at distance or over extended time periods. Without the need to be physically present, or to attend at the same time as other participants, group members may find that prior engagements are less likely to conflict with modelling activities. In principle, more people could participate, for longer. Computer-mediated facilities are known to improve the quality of contributions to group work, as inhibiting factors are reduced (Kiesler et al 1984). Cognitive loafing (Weldon and Gargano 1988) and social loafing (Latané et al 1979) may also be reduced if contributions to the group effort are solicited on an individual basis. Remote modelling would presumably tend also to discourage groupthink (Janis 1972) since it would allow participation by greater numbers, helping to introduce outside opinion, and would promote the review of decisions by the participants ('sleeping on it'). The need to complete modelling work within the fixed duration of a meeting is stressful and can have a negative impact on model quality (Holsti 1971). Relaxing this constraint would be likely to have positive effects (Burnstein and Vinokur 1975). In computer conferencing, where technology is used as the primary medium for group interaction, one of the main advantages cited is the ability of individuals to participate on an ad hoc basis (Kiesler et al 1984).

Principle 3.1	*Allow remote modelling at a distance and over time, by virtual groups.*

Good communication

A conventional (but naïve) view of information systems requirements analysis sees the analyst documenting known information or processing needs and the designer finding optimal (or as nearly optimal as possible) ways of satisfying the well-defined requirements. But analysis conducted in real life is never as straightforward (Bubenko 1986). Needs may not be known and, even if they can be stated, may be inappropriate. Requirements may be unachievable, impractical, or damaging. They may be based on unwarranted preconceptions. More often than not, there is conflict within the organisation about priorities or even about requirements.

A more realistic view of requirements gathering sees it as an exploratory process. Designers and users together learn and negotiate to achieve a useful end result in a practical time frame (McGinnes 1994). The process of negotiation and refinement of ideas demands powerful ways of capturing and conveying quickly-changing information. Visual presentation of information is one of the most effective ways of providing group members with a suitable problem-solving framework, and visual media such as white boards, flip charts, and projectors are widely employed for this purpose. In JAD, where good communication is emphasised and speed is a priority, it is suggested that magnetic symbols and CASE tools be used to visually convey information in a dynamic manner (Wood and Silver 1995).

Principle 3.2	Use visual aids to support group exploration, learning, negotiation and refinement

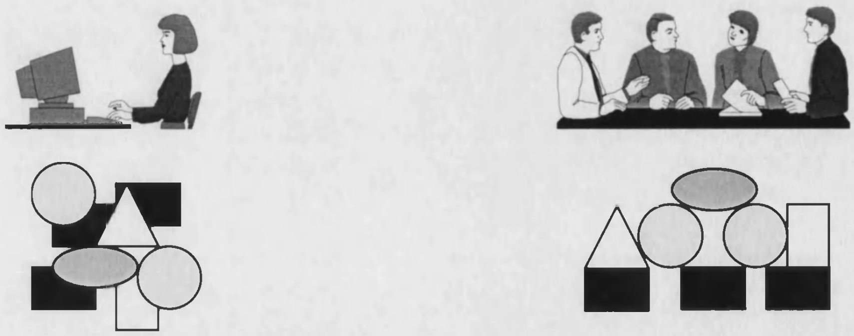
Presentation of information

It is known that groups perform less well in tasks requiring sustained and careful thought. Hence the need for clear and strong structure is especially true of group work. Research on problem-solving suggests that the ways in which information is presented can have a profound effect on the ability to solve problems, and can provide structure to a problem situation that would otherwise be missing (Bellezza 1983). Experts possess structured mental frameworks for thinking about problem situations and can organise material in their own areas of expertise according to these structures (Best 1999). A novice typically has not had the experience to develop such mental structures but, if equipped with a suitable framework for thinking about a problem, can emulate an expert's mode of problem-solving.

Principle 3.3	*Provide strong and clear model structure and frameworks to aid group comprehension and thinking.*

Group strengths

Research into group problem-solving and decision-making shows that groups tend to perform better than individuals in tasks that require generation of ideas or the selection of a solution from a set of alternatives (Steiner 1972). It is thought that the likelihood of the 'right' answer being suggested and the 'wrong' answers being weeded out increases with group size. But the distractions of group work militate against a sufficiently thorough treatment being afforded to problems that require sustained and deep thought. Hence individuals outperform groups at solving 'tough', in-depth problems. Groups such as committees are renowned for making ill-thought out and inappropriate design decisions. This implies that modelling techniques should be applied in different ways depending on whether modelling is being conducted by an individual or by a group. For group work, less emphasis must be placed on achieving correctness and more on generating ideas and finding alternative ways of representing them. This requires a less goal-oriented approach, with learning and helping emphasised in favour of solving problems and generating definitive results (Burnstein and Vinokur 1975). Since groups are more effective at selection than creation, it would make sense to allow them to build models by selecting and assembling predefined components. The hard thinking involved in creating reusable components is better done by individuals with specific expertise. This arrangement is illustrated in Figure 3.11 and summarised in Table 3.5. It is related to ongoing work on application construction through assembly of existing software components (Hofman 1995).



Model components created by individuals — Models created by groups

Figure 3.11 Individual and group modelling efforts

Existing conceptual modelling techniques permit this way of working only if used with some discipline. Data entities and object classes from existing diagrams can be reused, with new associations, in new diagrams. However, the level of detailed knowledge required to make this work 'on the fly' in a group setting would probably rule it out for groups not equipped with a CASE expert. Entities and object classes are not cleanly reusable since they do not encapsulate their own associations. It is meaningless to reuse a concept without reusing its associations to related concepts, since it is precisely the associations that serve to define it. To allow clean and quick reuse of concepts would require a way of defining concepts that *encapsulates* associations to other concepts.

	Individual/small group	**Large group**
Focus	Development of lower-level components and building blocks	Creation of models and higher-level components using building blocks.
Primary tasks	Invention Creation Construction	Selection Verification Combination

Table 3.5 Role of groups in modelling according to group size

Principle 3.4	*Allow groups to work with model components already created and verified by other modellers.*

Brainstorming

Trying to reach a unanimous decision in a group often causes difficulty. The most effective groups are prepared to accept satisfactory solutions rather than optimal ones. Group work tends to be more productive when the potential for criticism and competition between group members is lessened. If the aim of a session is seen as exploring a subject area and helping one another to learn, rather than reaching a unanimous position, participants are more likely to contribute in the required spirit (Zander 1989). The absence of judgement and editing in brainstorming sessions can encourage group members to make contributions and to suggest more radical and creative ideas (Osborn 1957). Individuals can also brainstorm very effectively on their own (Lamm and Trommsdorf 1973).

Conventional conceptual modelling techniques are unforgiving when used in a brainstorming context. During the early stages of modelling ideas are thrown around, and it can be counterproductive if they must be 'edited' before being recorded. Experimentation with alternative ideas is often needed. But most CASE tools are unable to represent ideas unless the ideas are re-expressed in the correct syntactical form. Moreover, once a model has been started it can be difficult to 'change tack' and restructure it along new lines. One needs to be able to create inconsistent or incomplete models, that evolve only gradually into correct models. But, with conventional tools, there is a penalty for not getting it right first time. To allow for the lack of rigour inherent in group work, conceptual modelling techniques should tolerate inconsistency and incompleteness, whilst still allowing technical rigour to be introduced later. They must support the process by which models are refined until they reach a satisfactory state. Modellers should be able to devote their attention to the problem at hand and not be forced to attend to the niceties of modelling, such as correcting syntax and resolving inconsistency, at inappropriate times.

Principle 3.5	*Support brainstorming by facilitating capture of informal or unstructured ideas and allowing easy model restructuring.*

Making mistakes

Modellers inevitably make mistakes. But some modelling techniques encourage modellers to make them. For example, any technique that incorporates built-in redundancy provides the opportunity for inconsistency. In object modelling an association between two classes is equivalent to the presence of a pointer (foreign key) in one of the classes. Inexperienced modellers often include pointer attributes and associations that contradict one another. In data flow diagrams, it is possible to draw a data flow from an external entity to a data store, even though this is always an error. In object models, aggregation may be used when association (i.e. a pointer to an object) is required. The novice modeller, lacking a mental framework for modelling, is unlikely even to recognise errors of this type without external assistance. Conceptual modelling techniques ought to be robust enough to ignore these errors, or should treat equivalent constructions interchangeably. Ideally, errors of this sort should be impossible, or the modelling techniques should automatically (or easily) highlight and resolve them.

There is always the possibility that modellers will create confused results, especially when groups of people collaborate to build large models. Common problems centre around naming and terminology. Model elements may have distinct meanings but be named identically. The same facts may be encoded several times in different ways, and identical elements may be modelled under distinct names or in different subject areas. Current conceptual modelling techniques provide little support for preventing (or even highlighting) errors like these. Greater opportunities for error and inconsistency exist when distinct modelling techniques are used in parallel, as is recommended in most systems development methods. Information expressed using a data flow diagram can contradict that contained in an object model. Often, the contradictions are not obvious and recognising them can be almost impossible without 'detective work' and the kind of insight that develops only with experience.

Principle 3.6	*Tolerate, and reduce the likelihood of, inconsistency and other simple modelling errors*

Comparing alternatives

Any model proceeds through successive versions before it is finished. Often, this means that alternative model structures need to be tried out before a satisfactory approach can be chosen. If several modellers work on related areas, they will inevitably overlap to some extent and their models must be compared before being merged. In both cases, it should be possible to compare, and possibly merge, models without necessarily resolving their inconsistencies. What this requires is the ability to perform model comparison and 'what-if' analysis without committing to changes. Modelling techniques should support the discussion and consideration of alternatives that is often required before inconsistencies can be resolved (Janis and Mann 1977). Some current CASE tools permit models to be merged but, in most cases, any inconsistencies must be resolved immediately. Models cannot be internally inconsistent. This places an unrealistic demand on the modeller since it is often not possible to solve problems at the time they come to light. Human organisations do not work to schedules demanded by software tools.

Principle 3.7	*Allow alternatives to be compared and explored before decisions are made*

Ensuring effective participation

The opinions expressed most clearly and loudly in a group will often be those of the dominant members. Apparent unanimity and agreement may be gained at the expense of commitment from other group members (Janis 1972). Research has shown that several factors influence how much an individual participates in a group. The presence of other group members can reduce one's own input through loafing. Inhibition can come about because of differences in status. The effects of both factors can be reduced by creating smaller, more homogenous groups and by avoiding face-to-face contact (Zander 1989).

One of the main ways in which active participation can be encouraged is by allowing ideas and suggestions to be explored without the necessity for commitment. Group members may be reluctant to contribute suggestions if they feel that there is a risk of wasting time or appearing foolish (Burnstein and Vinokur 1975). Allowing their ideas to be followed through, without the possibility of an adverse reaction should the ideas turn out not to be useful, can help to create an atmosphere in which group members are less likely to censor their own responses (Sheffield 1936). In modelling terms, this could mean making model construction so easy that creating and discarding models has little penalty, as in the use spreadsheets for 'what-if' analyses. Allowing individuals the time and facilities to explore their own ideas, before they are exposed to group opinion, would also provide the same sort of safety (Zander 1989).

The task of structuring a group for conceptual modelling, and planning its work, involves a trade-off between including enough people who have relevant knowledge, and keeping the group sufficiently small that useful results can be obtained. Research has shown that social and cognitive loafing increases with group size (Weldon and Gargano 1988, Latané et al 1979). But the mix of participants must be chosen so that the area covered by the meeting is of relevance to everyone present; otherwise, people will be unwilling to contribute and may feel that their time is being wasted (Avery and Baker 1990). The best approach seems to be to allow smaller groups (e.g. sub-committees) to work on specific areas, with the results being combined at a later stage through the action of a responsible discussion leader. This can be done with present-day tools but requires great discipline and a good deal of reconciliation when separate models are combined. It would be more practical if modelling techniques could support parallel work on the same model by more than one group. Small groups or individuals could work independently on model sections in their own areas of expertise, later bringing their work to group meetings where the ideas are combined and refined. This way of working is suggestive of the nominal group and Delphi techniques that have been found to increase the quantity and quality of participation in group work (Delbeq et al 1975, Rohlen 1975).

Principle 3.8	*Allow models to be constructed jointly by smaller groups and individuals.*

Arousal and attention

Participants in groups often need to maintain attention for sustained periods; in large groups it is easy to become bored and distracted, and concentration inevitably suffers. Traditional conceptual modelling sessions are almost guaranteed to result in lowered attention levels with their use of static, monochrome visuals and lack of action. For some users this presents an ideal opportunity to catch up on much-need sleep! It is in the modeller's interest to address these issues since individual performance in groups depends heavily on arousal and attention. Arousal is increased by colourful, animated surroundings and by the occurrence of novel events, all of which can be introduced deliberately. Increased personal involvement by group members, and frequent, enforced changes in task and posture can help ensure that attention is stimulated.

Principle 3.9	*Use all means available to retain attention and maintain arousal.*

Set

One of the most important factors affecting success in problem-solving is the range of characteristics known as *set* (Malim 1994). A set is a predisposition towards viewing a situation in one particular way. In the context of conceptual modelling techniques it is common for users to feel that models are 'technical' and the property of systems developers. The stated intention may be to capture relevant aspects of business reality, but users view models as relevant more to the analyst's work than their own (Bansler and Bødker 1993). This type of set has several origins. Users may be more interested in day-to-day work which has personal relevance to them. But they may also fail to understand what is being said. Computer professionals use specialised jargon at almost every opportunity, and conceptual modelling practice is no exception, with many terms and procedures foreign to the user's world. Even end-user orientated techniques like JAD introduce their own jargon (*facilitator, workshop, model, ...*). The result is often confusion. Analysts use inappropriate models in the mistaken belief that they reflect the business, while users 'go along with' models they understand only poorly. Personal experience certainly suggests that end-users are often willing to agree to, and even pay significant amounts of money for, the development of software despite not fully understanding what they will get. The technicality of models can be reduced by avoiding jargon (like 'object') in favour of the user's own concepts and terms, and by making models more recognisable (and thus intuitively understandable). It can also be decreased by ensuring that both the method and the purpose of modelling are immediately obvious to participants.

Another type of set that can have profound consequences for conceptual modellers is *set of function* (Malim 1994), the implicit assumption that the elements of a problem have a fixed function. Psychological research has shown that this type of set is a major reason for inability to find solutions to problems. Even when tools are present that could be used to resolve a situation, individuals may fail to recognise them as such if they must be used in unconventional ways. This is a problem in conceptual modelling. To successfully model complex situations often requires ingenuity. But novice modellers and modelling group participants may not recognise the full capabilities of the techniques or tools they use, and consequently experience difficulty in knowing how to represent particular situations. Training can help, but it is usually infeasible to train users adequately in conceptual modelling techniques, given the numbers that need to participate and the timescales, costs and other business pressures involved. The problem can be eased by making modelling techniques more accessible so that the function of model constructs becomes more obvious. As in user interface design, a well-designed modelling technique is one that needs least explanation (Nielsen 1994).

One way of ensuring that end users understand conceptual models is by making them responsible for their own models (Avery and Baker 1990). If suitable tools were available it is possible that some users could become proficient conceptual modellers without the need for extensive training or assistance. However, for this to become a practical prospect, modelling would have to be made as easy as, say, creating a simple spreadsheet. The risk of loafing would certainly be reduced if model production were the responsibility of each group member individually. This idea would also require a change in set by IT professionals, who may view modelling as a technical process over which they must retain control. Modelling would have to be seen as a user-controlled process in which the user's own concepts and terminology were paramount.

Principle 3.10	*Give users control and reduce unhelpful sets by avoiding jargon and making both the purpose and method of modelling intuitively obvious.*

3.5 An alternative approach to conceptual modelling

In this chapter we have presented a series of principles that can be applied to modelling techniques. The challenge is apply these principles coherently in a technique that is useful. Below we present method 'X', a conceptual modelling technique that conforms to the psychological principles. Method 'X' is only one out of many possible techniques. The principles are used to explain the modelling technique. The narrow of the two definitions of conceptual modelling (see Chapter 1) is used here, for practical reasons, because it is, by definition, the more specific of the two. A narrower focus on specific key aspects of conceptual modelling practice allows a more tightly-focused experiment.

Modelling concepts

Principle 1.2 states that conceptual models should be like mental models. A mental model is a visualisation which illustrates one particular way in which a situation could occur, but which can symbolise all occurrences of the situation. Using a theatrical analogy, we could think of a mental model as a scene within which the actors participating in a situation act out their roles. The scene consists of fixed and moveable elements representing buildings, equipment and furniture. The actors are people and the organisations they represent. The scene takes places against a backdrop that symbolises physical location. Within the scene certain less concrete concepts are represented, symbolising intangibles such as ideas, information and intentions. As the action proceeds, elements of the scene or the location of the scene may change. A mental model may involve many individual scenes, strung together to form a narrative (Laurel 1993).

We can define a conceptual model as a collection of related components that stand in for mental concepts. The most fundamental question is what kind of components a conceptual model should include. We know that they should represent concrete business concepts (Principles 2.1 and 2.2), which exist in the user's world (Principle 2.4) and are named according to the user's own vocabulary (Principle 1.3). They should be pitched at an everyday level of generality (Principle 2.8). The set of available concepts should be rich and extensible (Principle 2.4) and the requirement for simplicity demands that there be a one-to-one correspondence between model components and mental concepts (Principle 2.9). Method 'X' assumes that the user's world is a business situation of some form and uses the set of concept types given in Table 3.6. The range of mental concepts that can be modelled using this set of concept types is unbounded, provided that each falls within one of the types. The types are straightforward and appear to cover most business concepts reasonably naturally. One might expect business users to grasp them quickly. One potentially problematic type is 'conceptual object', which may not be pitched at an everyday level of generality for most users, who would probably be unused to thinking about conceptual objects in a single category. However, the range of potential conceptual objects is so vast that it is impractical to use a large set of predefined alternatives.

If we are interested in capturing information requirements as part of the conceptual model then base-level data types (date/time, text and number) should be included so that they can be assigned to the higher-level business constructs. In addition, unstructured information types need to be included so that early, unstructured ideas and model annotations can be incorporated (Principle 2.11).

Unstructured information types could include notes, pictures and embedded document or hypertext links. Table 3.7 brings together the set of modelling constructs supported by method 'X'.

Concept type	Represents	Example
Person	Individual person	Claims adjuster
Organisation	Identifiable group of people	Insurance company
Activity	Event, process or activity; something that happens	Purchase
Place	Physical location	Supermarket
Physical object	Concrete, physical object	Car
Document	Information on paper or in electronic form	Bank statement
Category	Way of grouping or classifying things	Gender
Conceptual object	Idea, information in abstract form	Law
System	Technology such as computer-based information system	Payroll system

Table 3.6 Method 'X' component types

<table>
<tr><td rowspan="3">All model components</td><td rowspan="2">Structured model components</td><td>Business concepts</td><td>Person
Organisation
Document
Conceptual object
Physical object
Place
Category
Activity
System</td></tr>
<tr><td>Data items</td><td>Text
Date/time
Number</td></tr>
<tr><td>Unstructured model components</td><td>Annotations</td><td>Notes
Picture
Embedded/linked document
Hypertext link</td></tr>
</table>

Table 3.7 Taxonomy of method 'X' modelling constructs

Describing or defining concepts

The most fundamental building blocks of conceptual models are model components that stand in for mental concepts. Each mental concept has significance formed by associations to related concepts and sensations that together constitute its meaning. Without these associations there would be no meaning. Similarly, a model component lacks meaning if it is not adequately explained or if the viewer cannot understand the explanation. By extension a conceptual model can be considered meaningless if the components used to express it have no meaning themselves.

How can we ensure that the meaning of model components can be conveyed adequately? Most importantly, it should be simple to define concepts without making mistakes (Principle 3.6) and any definition should be easily understandable (Principle 1.1), employing the user's own terms in a structured way (Principle 1.3). Principle 2.7 states that alternative methods of defining concepts should be offered. The ability to use semantic associative retrieval (Principle 2.10) requires that concepts be defined by reference to other concepts, and it must be possible to be able to construct components from other prefabricated components (Principle 3.4). It should also be possible to define concepts in an unstructured way (Principle 2.11), and pictures could be used together with words where they help to make meaning clear (Principles 1.4 and 1.5). To maximise understanding, a clear and strong structure is beneficial (Principle 3.3) and short-term memory can be aided by chunking methods such as partitioning, hierarchical organisation, decomposition, categorisation and aggregation (Principle 2.6). In method 'X' model components representing business concepts are described in three ways: by naming, by annotation and by association to other known concepts.

Naming In a mental model, concepts define themselves and need not necessarily be named. However, for our purposes it is necessary to insist that each concept has a unique name, if only for practical model management reasons. The name represents the role that the concept plays in a model. For example, in a model connected with student enrolments, a component of type *person* might appear as *student*, whereas it could appear as *employee* in another model. As in everyday subjective thinking, there is no absolute distinction between component types and roles. A concept named student may appear as *scholarship applicant* elsewhere in a model. The connection between a concept and the roles that it plays is a useful form of chunking based on categorisation. Both the concept itself and the more restrictive role should be named using familiar and meaningful terminology.

Annotation Each component can be annotated or described to help make its meaning clear, using text, pictures, references to documents, hyperlinks and so on, to create a 'collage' of meaning. An unstructured concept definition may consist solely of one or more annotations. Annotations may be refined and may eventually give rise to one or more structured model components.

Association Concepts may be described in terms of other concepts. For example, the characteristics of a car include its colour, its price, its year of manufacture, the fact that it is a vehicle, and so on. We can think of the concept *car* being defined with reference to these other concepts *colour, price, year, vehicle*. The other concepts may have to be defined or may be prefabricated.

Representing each concept as a 'chunk' of knowledge in this way provides another useful chunking scheme for models (based on aggregation or decomposition). The model is presented to the user in hierarchical form (see below).

One special case is that of activity components, which represent procedures or business processes. They can be described as sequences of steps which together constitute a scene (analogous to a mental model or situational memory). Specific model concepts participate in each step much as the actors in a play obey stage directions in a script. The activity may be enacted using simple animation techniques, exploiting visual imagery to help create more understandable models (Santamaria 1999).

Combining concepts in models

We have said that we may define concepts in terms of their relationships to other concepts. If we are interested in capturing the nuances of real life then the number of ways in which concepts may be related is enormous. But perhaps, taking what we know about brain function into account, there are fewer ways in which concepts need to be related, since in the mind we could say that any two concepts are either associated or they are not, to a greater or lesser degree. To ensure simplicity, structure and understandability (Principles 1.1, 2.9, 3.3) model concepts may be related in method 'X' in only two ways: *contains* and *like*. The *contains* relationship identifies constituent parts of a concept. For example, in a model component representing a car, the constituent parts might represent wheels and doors (plural), an engine (singular), and so on. Constituent components are made singular or plural to represent either a single instance of a concept, or many instances. Although this example is based on physical inclusion, conceptual inclusion is also possible (e.g. the concept *car* includes the concept *manufacturer* even though cars do not contain car manufacturers in reality). The *like* relationship indicates that a concept is similar (or equivalent) to another concept. For example, when the concept *person* is used in a model with role *employee*, then we can assume that employees are or are the same as people. The concept *employee* is not identical to the concept *person* since it relates to fewer individuals (if we treat *person* as the set of all people, then the set *employee* is a subset of *person*).

Some modelling techniques distinguish more finely between different types of relationship. In particular, the two types of inclusion known as aggregation and association are usually distinguished from one another. The constituents of a real-world object may represent things temporarily included or shared between several other objects. The pilot of an aircraft cannot be considered as physically part of the aircraft and is likely to fly in several different aircraft. But it is useful to think of the concept pilot as included in the general concept of an aircraft. When a model is constructed it may be unclear whether to use aggregation or association. Over time an aggregated component may turn out to be an associated one. Using the car example, an engine which is permanently attached to a car (aggregation) could at a later stage be moved to a different car (association). An analogous argument applies to the distinction between attributes and aggregation. What are thought of simply as attributes may turn out to represent full-blown concepts. A car manufacturer may initially be recorded as the attribute manufacturer name only to become the concept car manufacturer at a later stage. Since these are fine distinctions, arguably better made at the system design stage, the modelling technique ignores them.

Support for multiple models

It is inevitable that conceptual models must be compared and combined (Principles 3.4, 3.7, 3.8) especially if prepared in a group context. Like mental models, different individuals' conceptual models of the same situation may differ, and the definition of any given concept may vary from model to model. Dividing a business area into models is itself a chunking strategy that is based less on meaning than on practicality. A model is an arbitrary grouping of components that need not necessarily form a coherent whole, but it is convenient to keep possibly related concepts together in the form of a model for administrative purposes. To provide further administrative convenience method 'X' incorporates an additional structuring construct: *the model file*. Models are collected together in files to provide a component base from which new models can be constructed. The different models in a file might represent distinct functional areas of an organisation, alternative views of the same functional area, or different versions of a single view. The need to avoid premature attempts at consensus means that models within the same model file could contradict one another in one or more ways. In view of this, concepts will not automatically be shared between models. Although possibly useful existing concepts might be suggested, the act of reusing an existing concept must be a conscious one, to avoid confusion.

Appearance of constructs

The appearance of model constructs must help to make them understandable and unambiguous (Principle 1.1). One of the main goals is to take advantage of unconscious or pre-attentive processing (Principle 1.2), which relies on the use of highly recognisable and distinctive visual symbols. Pre-attentive processing can be extremely efficient (Principle 1.4), especially when familiar words and pictures are used in conjunction to provide reinforcement and context (Principles 1.3, 1.5, 1.8). As in graphic design, the optimum symbol is one that conveys the maximum information with the minimum conscious effort by the viewer (Principle 1.5). Economy of representation is desirable both for ease of implementation and to avoid information overload. Symbols should be recognisable visual representations of the things they represent. But symbols that are too specific can lose visual economy and may convey unintended meaning. They must be specific enough to represent real things but general enough to stand in for generic concepts (Principle 1.9). Symbols must be used consistently (Principle 1.10) and only one symbol should denote each business concept (Principle 2.9). Ideally, users should choose symbols themselves, to maximise personal relevance (Principle 2.5). Method 'X' uses symbols for the business concept types as shown in Figure 3.12.

Figure 3.12 Method 'X' symbols for concepts

The user can search for alternative symbols, to denote more specific concepts, so that each distinct concept may be represented unambiguously. If no suitable symbol is readily available then it is possible to quickly and easily creating new ones. The ability to combine existing symbols is especially useful for compound concepts. For example, an airport could be represented by combining symbols for a building and an aircraft, just as many English words, especially those of Scandinavian origin, have been formed by combining existing words (Burgess 1992). The word 'airport' itself is one example. When the symbols are used within a model the symbol label gives the specific concept name (or role, if different). If a concept is plural, three dots are placed next to its symbol and its name is made plural. If a concept is optional, its name is enclosed in parentheses, as shown in figure 3.13 (note the compound symbol for 'flight' created by combining two separate symbols).

Figure 3.13 Plural and optional concepts

Standard symbols are also used for the three kinds of data item and four types of annotation, as shown in Figure 3.14.

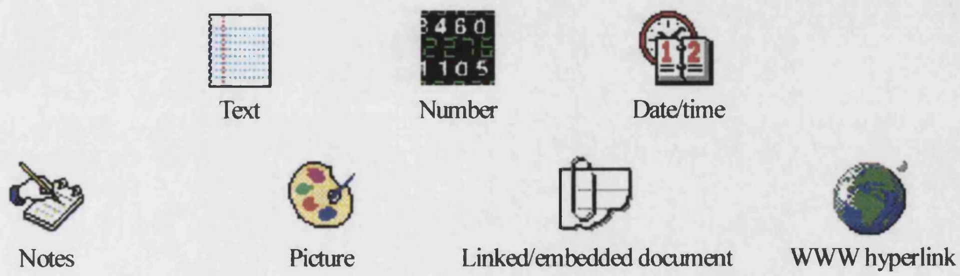


Figure 3.14 Method 'X' symbols for data item types and annotation types

The standard symbols shown here are all icon-sized graphics, which is necessary because of the use of buttons in the tool's user interface (see Figure 3.16). However, larger graphics are just as effective, if not more so, and therefore the library of images included in the tool contains many larger images.

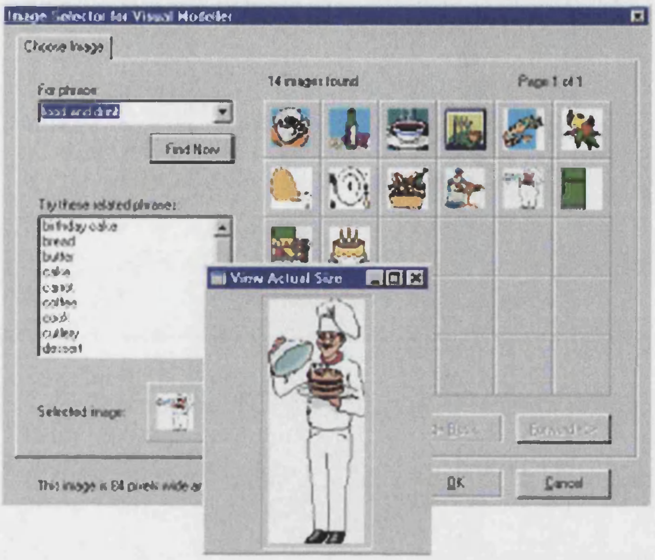


Figure 3.15 Searching for an image

Use of image annotations

Image annotations may be used to illustrate aspects of a model. For example, a scanned image of a document can be incorporated to illustrate the document's visual appearance. As with all other types of model information, document layouts can be used as the basis for deriving new or more detailed model components. The image itself can be annotated so that specific parts or aspects are highlighted, perhaps to indicate sub-components of the document's structure. Images may also be particularly useful to illustrate activities in progress or to provide a backdrop against which activities can take place. For example, an activity carried out by a bank might be depicted against a backdrop illustrating the interior of a bank branch (Laurel 1993). Video images could also help illuminate activities of a sufficiently observable nature (O'Donnell 1997). Assembling a collection of annotations to help explain a model and to present relevant issues is analogous to making a collage—gluing together any useful pieces that help to create the overall picture. This is an obvious application for multimedia technology, but relies on the availability of suitable source material and the ability to find and use it 'on the fly' in modelling sessions.

Appearance of models

The visual appearance of models can be an important aid to groups in learning, negotiation and the exploration and refinement of models. Principle 1.2 states that conceptual models should resemble both the situations they describe and mental models of the same situations. Visual chunking methods (Principle 2.6) and layout should be used to improve understandability (Principle 1.6) and provide context (Principle 1.8). Preattentive (unconscious) processing should be encouraged. The representation used in method 'X' exploits the familiar 'graphical user interface' style, since this is well-known to many end users. Model concepts are represented by icons in windows, and can be exploded (by double-clicking) to reveal further windows containing icons that represent associated concepts. This is a simple but powerful way of expressing the 'contains' relationship between concepts through a visual analogy (the window contains the icons). Icons can be created, moved, copied and deleted in familiar ways, to represent the manipulation of concepts in a model. Since many users understand how to manipulate icons and windows, they already know a good deal about how to manipulate concepts in models, and no new notation needs to be learned.

It is worth noting that most existing CASE tools do not exploit already-known modes of interaction, and force the user to learn new and distinct procedures. In some tools the distinction between user interface and modelling technique is stark, and the methods for editing a model are quite divorced from normal ways of working with a graphical user interface (e.g. Popkin 1995, LBMS 1993). Possibly, this is a consequence of the fact that the modelling techniques developed many years before graphical user interfaces became available, although that does not explain why even recent techniques suffer from the same defect.

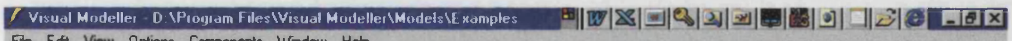


Figure 3.16 Method 'X' model appearance

Process of model development

The process of model development is aimed at organising and refining the ideas expressed in a model until sufficient structure and coherence have been achieved. For the smallest systems, individuals may work alone on this task. But for most non-trivial systems, a group effort is needed. Groups are most effective at selection and evaluation tasks and least effective when careful and sustained cognitive effort are called for. Hence support must be given if groups are to perform adequately in modelling work. Most importantly, the purpose and method of modelling should be intuitively obvious, and jargon should be avoided (Principle 3.10). The scope for errors should be reduced (Principle 3.6). Clear structure must be provided to guide thinking (Principle 3.3) and visual aids should be used (Principle 3.2). Attention and arousal must be maintained (Principle 3.9). It can help for groups to work with prefabricated model components (Principle 3.4) and to extend their collaborations over time and between locations (Principle 3.1) combining work by individuals, small groups and larger groups (Principle 3.8). Brainstorming techniques can help groups work more effectively (Principle 3.5) and additional support for comparison and exploration of models is useful (Principle 3.7). The method 'X' modelling process is illustrated in Figures 3.17 to 3.19. In this example, the starting point is an informal narrative description. However, the process is designed to work equally well with an initial brainstorming stage to generate a list of possible model concepts, or even if based on formal pre-existing documentation.

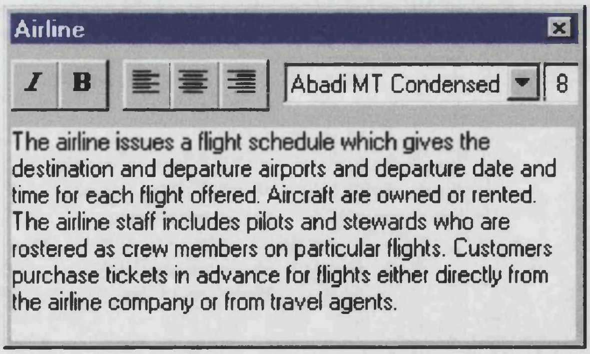


Figure 3.17 Initial notes

The modeller creates new components in the model to represent relevant concepts that have been identified from the narrative. The position of the various elements within the model is left up to the modeller, so that a meaningful layout can be created. The model is developed by addition of further symbols, representing business concepts, annotations and data items. Structure is added to the model by placing symbols inside windows, to represent relationships between concepts, and by making concepts optional or plural as appropriate. The ability to annotate models helps to ensure that all important information is captured, even if it cannot be expressed in a structured form.

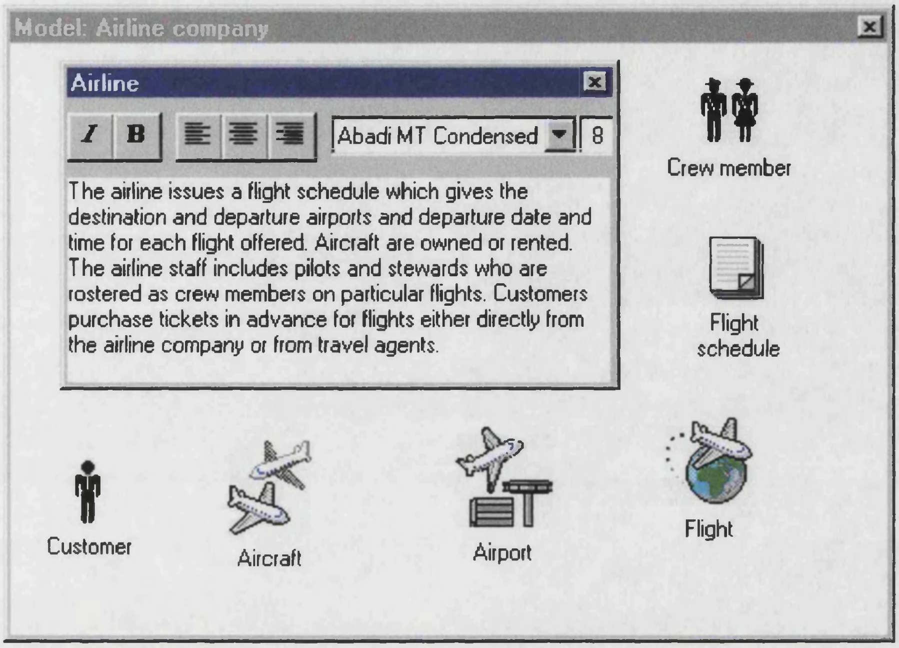


Figure 3.18 Beginning to identify model concepts

The model shown above in Figure 3.19 has meaning on more than one level. It can be interpreted impressionistically in the same way that one might interpret a multimedia presentation. But it also has the formal meaning outlined in Table 3.8.

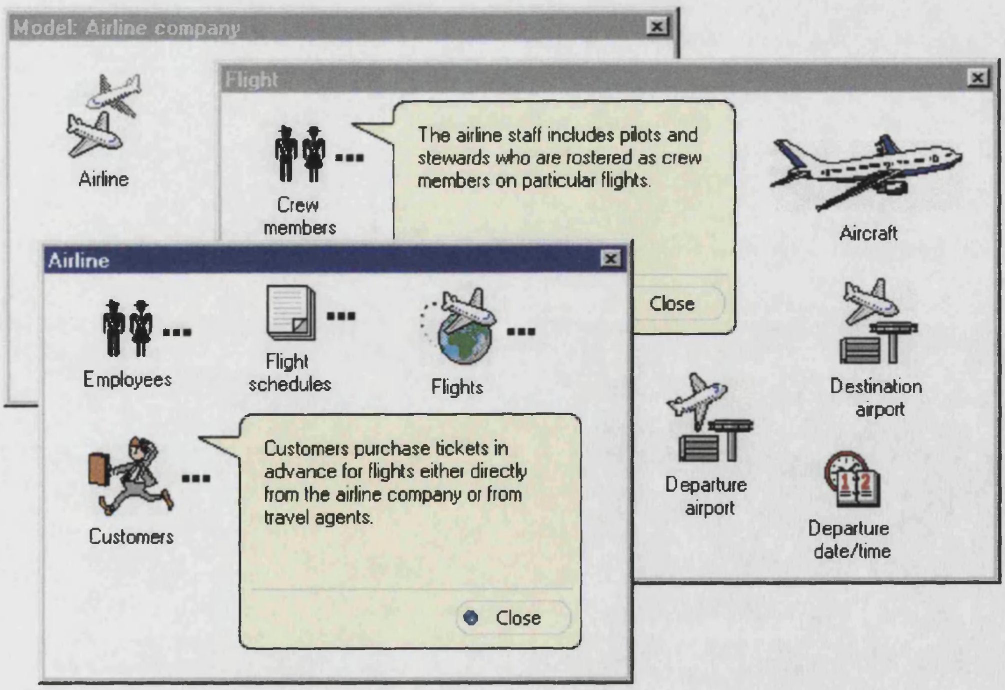


Figure 3.19 Structuring the model

Each airline has one or more employees (which are people), one or more flight schedules (documents), flights (activities) and customers (people). A flight has one or more crew members (which are airline employees), departure and destination airports (places), an aircraft (physical object) and a date and time of departure.

Table 3.8 Formal meaning of model

We now turn to the process of modelling. Systems development was for a long time seen as a linear sequence of steps. This view has now given way to a more iterative model in which the steps are to some extent merged and overlapping. Prototyping and RAD methods such as DSDM (Stapleton 1997) have helped to blur the distinctions between analysis, design and programming. The method 'X' process is no exception. Table 3.9 presents one view of the 'traditional' life cycle alongside the method 'X' process. The starting point for both is that an appropriately-qualified analyst is presented with a brief to investigate requirements in a particular functional area.

Stage	**Traditional approach**	**Method 'X'**
Preparation	Business analyst gathers and reviews available background information and identifies relevant users.	Analyst gathers and reviews available background information and identifies relevant users.
Analysis	Business analyst interviews users. Business analyst documents his/her own understanding of requirements in a User Requirements Specification document. Users are requested to read and verify the User Requirements Specification. Changes are made if necessary prior to formal agreement.	Analyst prepares modelling tool with suitable symbols, images, prior models, etc. Analyst conducts modelling sessions with users and selected users refine models in own time. Focused groups compare models and produce definitive versions. Joint authors of models generate a hardcopy version of the models for formal acceptance by management.
Design	Systems analyst reinterprets the User Requirements Specification to create logical models (e.g. object models, dataflow diagrams). Users may be requested to sign off logical models. Designer reinterprets logical models to create program and database designs.	Logical models and system designs are derived directly from user models.

Table 3.9 Possible approaches to requirements analysis and design

In method 'X, model development is a collaborative activity in which users and analyst work together to create and refine models of their own organisations and business areas. This is very close to the JAD approach (Wood and Silver 1995). The analyst acts as a facilitator. Because the models are produced by end users, the processes surrounding verification and agreement are different. The 'signing-off' process is less important. There is also less need to translate mentally or cross-check between prose requirements, data models, process specifications and other representations; the perspectives underlying these representations are alternative views of a single model.

Preparation for modelling

To model an organisation or business area accurately requires careful thought. In order to get the best out of group work, aspects of the work that require excessive cognitive effort should be reserved for individuals to do. In particular, the analyst must make full preparation for modelling. The objectives (e.g. creation of new or improved information systems, or business process re-engineering) must be agreed. The scope of the exercise must also be established, identifying the parts of the business that are of interest. Users with relevant knowledge, who are willing and able to participate, must be selected. Suitable users normally include those with day-to-day operational responsibilities, as well as managers who have a high-level but perhaps less detailed view (Wood and Silver 1995). To help reduce inhibition, however, groups should be arranged so that disparity in status between participants is minimised.

The analyst must read available material on the relevant business area(s) and then gather any existing models in related areas. The existing models are useful because they can provide context and give valuable insight to the participants. Existing material may be a suitable starting point for modelling or, if not sufficiently relevant, it may be useful as a source of components for the construction of new models. The analyst must also represent concepts distinctively using graphical symbols chosen from the wide variety of existing sources for images (e.g. CD-ROMs and the Internet), or at least ensure that suitable symbols are available for later use.

Reference material that may support modelling (such as memoranda, strategic plans, requirements statements, proposals, specifications, and tenders) can obtained for inclusion in models as annotations. Paper document such as forms can be scanned electronically. Other relevant text can be made available in electronic form. Most organisations have a wide range of such documents, and replicating their content may introduce undesirable version control problems. Hence the analyst must have the choice of referencing existing material stored elsewhere (e.g. by linking, bookmarks or shortcuts to documents held on a corporate intranet) as well as including material directly in models (e.g. by document embedding or cutting and pasting).

Modelling sessions

Selected users are brought together in focused groups, each with a specific remit to model a given part of the overall business area. Key users may participate in more than one group. The objectives and scope of the exercise must be understood by group members, but the analyst must also take time to discover what the group members themselves would like to get out of the exercise, and how they feel the results could be of use. Obtaining their commitment is essential at this point, and it can help a good deal if group members feel they have something to offer. Group members are introduced to the modelling technique, perhaps by viewing a simple demonstration model or a relevant existing model. The use of attractive and colourful images, animation and sound can help capture attention and generate enthusiasm at this stage.

The analyst facilitates the modelling process by prompting users for information in a neutral way, requesting, for example, that a particular user talk through a process he or she knows well. The analyst must steer the overall process and keep discussion on relevant topics, demonstrating how group members' business knowledge may be translated into models. In the absence of existing models, key words or phrases may be recorded, forming a list of potential model components as a starting point. The group can then choose an appropriate type and symbol for each item. In situations where some modelling has already been done in related functional areas, existing model components can be examined and used if suitable.

In order to foster commitment and enthusiasm it is advisable for group members to be involved actively (Lawson and LaFasto 1989). Figures 3.17-19 illustrate a process driven initially by analysis of some pre-existing text, largely because this can be shown easily. But there can be benefits in having group members say in their own words how they perceive a business area to be structured rather than passively dealing with material someone else has produced. The text would then be useful more as model annotation and for later comparison. The group members' contributions may be recorded for later analysis.

Groups may convene repeatedly until a satisfactory picture of the relevant aspects of the organisation has evolved. To aid memory, the model should look at the start of each session as it did at the end of the previous session. Between sessions, individual members may work on models in their own time and their models may be explored and refined by the group or may simply provide a focus for discussion of alternatives. If work is proceeding in several groups, the analyst may need to integrate or compare models, and negotiation may be required between groups to achieve consensus. The analyst should act conservatively by preserving differences until they have genuinely been resolved. To impose consensus is inadvisable.

Role of supporting tool

A commercial software product has been developed to support the method 'X' modelling process. It offers a mechanism by which strong and clear structure is provided (Principle 3.3) to modelling efforts. The modelling tool's user interface contains a number of rather large, labelled buttons with pictures, which correspond to the building blocks of models: business concepts, annotations and data items. When modelling from scratch the user is prompted to begin correctly, by clicking on the buttons. The model components themselves resemble icons and so it is reasonably obvious for someone used to graphical user interfaces that one needs to click on them. This results in a window being opened, into which newly-defined components will be placed. In this way the tool naturally leads the modeller to create structure within the model from the earliest stages of modelling. The fact that related ideas are likely to be modelled in succession means that the assumed structure stands a good chance of being approximately correct, by default, without the need for conscious effort by the modeller.

Many aspects of model construction and manipulation discussed in this chapter would be infeasible without the use of a suitable software tool for modelling. For example, it would be impractical to conduct modelling sessions as described using whiteboard, flipchart, overhead projector foils or other visual aids alone. This is because of the dynamic nature of the method 'X' process and the need to manipulate large amounts of information quickly and easily (e.g. when reusing existing model components or when comparing models). It also helps satisfy the requirement for colour, movement and sound (e.g. video). In particular, it must be possible to revise models and to undo changes repeatedly, quickly and easily, *without penalty.* Any overhead in revising a model is likely to deter modellers from making corrections. The modelling tool must support the process of modelling in a natural way, not interfere with it. To meet the challenge of creating and refining models in a group setting, and to avoid the pitfalls of existing CASE tools (Connell and Shafer 1995) the goal in designing the tool was to exceed the ease of use and capabilities of conventional methods. A well-designed tool is invisible, in the sense that the user can forget they are using it and concentrate on the task at hand (Nielsen 1994). The tool's user interface and the modelling technique's own representation have therefore been made identical as far as possible (in other words, constructing a model consists largely of manipulating windows and icons).

The tool offers the modeller a virtually unlimited and reliable long-term memory for large and complex models, and allows this information to be retrieved by association, in some cases much as we recall information from our own long-term memories through cued recall (Avery and Baker 1990). For example, the tool will search for suitable images for components, based on subject matter, using a semantic association method. Over 1,500 images are available for immediate use covering a wide range of topics. A less highly selective form of retrieval is also available for model concepts, based on their type.

The tool offers a range of alternative model views (see Table 3.10). The window-icon and tree views would be familiar to most computer users. Both of these views present the model to the user as if it were hierarchical in nature. In fact a conceptual model is not a hierarchy but a network of linked concepts. It is thought that experts possess hierarchical mental frameworks that help them resolve complex situations. Non-experts typically do not have such frameworks, and external aids that provide suitable frameworks are beneficial. Presenting a model in a hierarchical way offers this benefit. In addition, an animated view (in construction) is available for activity components, and a prototype 3-D viewer (also under construction) allows models to be navigated in three-dimensional space. The mind is adept at perceiving information arranged in three-dimensional space (Herndon, van Dam and Gleicher 1994). Three-dimensional representations offer significant advantages since they permit more information to be presented and allow the focus of attention to shift over a greater (subjective) area. Users may also paste background images into model windows, helping to provide further context. Prototype view drivers (in construction) for the modelling tool allow models to be viewed in some 'traditional' formats, including object model (OMT), entity-relationship diagram, functional decomposition, flowchart, RAD diagram, DFD and use case diagram forms. The fact that individual events can be specified suggests that skeleton state-transition diagrams could also be produced, with a separate diagram for each model component. This has not been attempted, however. Data may also be exported in a form suitable for input to CASE tools so that their diagramming capabilities can also be used.

The tool is designed to work on standard IBM-compatible personal computers under Microsoft Windows 95 or 98. The platform was chosen for pragmatic reasons. Windows is well-known amongst business users, including people who would not class themselves as computer specialists. Great care has been taken to provide a modelling environment which is familiar and understandable to untrained end users. The look and feel of the tool is similar to other Windows applications, and this is intended to make the learning process and subsequent individual work easier. The interface is unlike that offered by some existing CASE tools, in which the user is presented with a series of forms in which to enter data about models. Interaction with the tool uses direct manipulation, where the user makes changes to models by altering the representation of the model on the screen. The mental connection between screen representation and model is carefully maintained so that users will identify the visual aspects of the model strongly with the model itself. Little is hidden.

Application of the tool in a group setting is enabled by the use of a projector and remote mouse or other pointing device. This allows the whole group to see the model at all times and to follow the model's development as new components are incorporated and existing structures amended. The aim is to replicate the ease of use of a whiteboard or flipchart whilst providing greater functionality. A colourful, animated style of presentation is intended to help retain interest and bolster arousal. A prototype 'remote modelling' feature is also being developed so that virtual groups can participate in modelling using the World-Wide Web. A 'Save As HTML' feature allows static copies of models to be published on web sites for perusal by extended groups.

To ensure understanding, a quick English-language summary of the model's meaning can be obtained, either for the whole model or a selected portion. The tool deduces relationships wherever possible using the most likely (statistically) cardinalities, but the user is free to amend the deduced relationships if necessary. The ability to deduce relationships based on partial model contents is analogous to psychological set or the effects of context, since the tool infers details about specific concepts based on their surroundings. The tool can also quickly generate new applications software, based on the current model, so that a modeller can experiment with different model configurations and view the resulting changes to screen design or database table structure within seconds (this feature in particular was found to be crucial in helping untrained end-users to model effectively, as outlined in Chapter 6). Users typically wish to experiment with their prototypes by entering data, and so a prototype data migrator tool is being developed that will automatically transfer previously-entered data into newly-generated applications, restructuring the data if necessary.

Maintenance accounts for a large proportion of the effort expended in systems development. Accordingly the tool will create models automatically by analysing the structure of existing database systems. Models created in this way can offer a very understandable business view of the data locked in existing systems, that may otherwise be lacking. After restructuring the model, new applications can be generated that automatically take the data from the existing systems. Table 3.10 summarises the features offered by the tool (note that only a subset of the features listed in the table were used in the later parts of this study).

Function	**Capabilities**	**Used in experiment?**
Model manipulation	Creating, saving, retrieving, copying models, etc. Comparing and merging models Reverse engineering existing database systems Importing/exporting to and from CASE tool, text files, etc. Checking model completeness	Yes No No No Yes
Model editing	Copy, cut and paste, drag-and-drop editing Automatically-deduced relationships OLE linking/embedding for docs, graphics, sound, executables, etc.	Yes Yes Yes
Graphical symbol functions	Automatic associative image search (1,500 images) Importing images	Yes Yes
Alternative views	Normal view (icons)/Tree view (hierarchy)/Object model (OMT) view Save as HTML (static) English language model interpretation (component or model)	Yes No Yes
Documentation	Document generation (rich text format) Print preview/print	No No
Application generation	Preview application user interface Generate full working applications in Microsoft Access Automatic data migration to new applications	Yes No No

Table 3.10 Features of modelling tool

Comparison with other techniques

The information contained in method 'X' models is related to that captured by other modelling techniques. Some examples of how the contents of models can be mapped between techniques are given in Table 3.11.

Modelling technique	Construct	Can be derived from Method 'X' elements
Object modelling/ Entity-relationship modelling	Classes/ Entity types	Business concepts (e.g. concepts of type person, organisation, document, etc.) Business concept types used in the model (e.g. the concept 'Person').
	Properties/ Attribute types	Data items (e.g. text, number and date/time).
	Associations/ Relationships	Associations between business concepts.
	Inheritance/ Subtype relationships	Implied relationships between any non-annotation model components and their types (e.g. employee—person). Roles (e.g. employee—contract employee).
	Cardinalities	Optionality (optional = minimum cardinality of zero, non-optional = minimum cardinality of one). Multiple (multiple = maximum cardinality many, non-multiple = maximum cardinality one).
	Operations/ Methods	Business concepts of type activity.
Data flow diagrams	External entities	Business concepts of type person, organisation, system
	Data stores	Business concepts containing data items
	Data elements	Data items
	Data flows	Business concepts of type document and conceptual object (the source and destination for each data flow can be deduced accurately only if the 'process animation' view is used to depict each activity in action).
	Processes	Business concepts of type activity.

Table 3.11 Comparison with selected conventional modelling techniques

Soft Systems Methodology (SSM) includes *rich pictures,* a diagrammatic techniques used to represent problem situations (see Appendix A). SSM rich pictures are often used to depict the parties involved in and subject matter of business problems. They can represent factors such as the influence of one group on another, areas of conflict, and physical inputs and outputs, as well as a range of other concerns. Because they use 'stick people' and recognisable symbols (such as crossed swords to depict conflict) rich pictures may be seen as similar to method 'X'. However, the two techniques are intended for quite different purposes. Method 'X' is essentially a means of defining business concepts, as a basis for systems design. Rich pictures and the other techniques in SSM are used in a much more general way to depict business problem situations, which is why SSM has been used as an 'early stage' method for several well-known information systems methods including Multiview (Avison, Wood-Harper, Vidgen and Wood 1996) and SSADM (Bryant 1995). Below we consider several key differences between method 'X' and SSM.

Each component of a method 'X' model refers to a specific business concept. Placing a component in a method 'X' model implies that the intention is to construct an information system that will store information *about the corresponding business concept.* SSM rich pictures are not intended for this purpose. For example, one may place a symbol in a rich picture to represent an issue such as 'quality', even if there is no intention of storing information about 'quality', or even creating an information system at all. SSM rich pictures incorporate no means of stating what concepts are to be implemented and what items of information are to be held for each concept.

Method 'X' models are intended to be capable of two types of interpretation: informal and formal. An untrained end user can interpret a method 'X' model informally but still gain a reasonably accurate idea of its meaning. A trained modeller (or a computer) can interpret a method 'X' model formally and produce a 100% accurate and unambiguous statement of its meaning, in the sense that a unique database design can reliably and consistently be constructed from the model by applying an entirely automatic process. The idea behind method 'X' is that there is some convergence between the informal and formal interpretations so that the novice can, to some extent, emulate the expert. In contrast, SSM rich pictures do not have distinct formal and informal interpretations. All interpretations of rich pictures are essentially informal, because the modeller is free to use the notation in a very flexible way, and one may interpret any diagram as one sees fit (perhaps the only 'true' interpretation is in the mind of the modeller). It is therefore not possible to construct an algorithm that can be guaranteed to convert an arbitrary rich picture into a useful database design.

Method 'X' models are intended to be similar to analogical mental models. Each business concept in a method 'X' model must represent a person, an organisation, a document, or one of the other predefined concrete concept types. Similarly, any given concept in a method 'X' model may be exploded to reveal its meaning in more detail, through its connections to other concepts. This is intended to emulate the way the mind creates meaning by linking ideas semantically. SSM rich pictures are not intended to correspond so closely to analogical mental models, if at all, and do not allow 'semantic' concept definition. They offer no formal means of denoting the meaning of a concept in terms of its relationships to other concepts. In fact, the symbols in rich pictures in many cases do not correspond to concrete entities at all, but to abstract ideas. There is no overarching set of concrete concept types from which the modeller must choose.

Other modelling techniques also bear certain superficial similarities to method 'X'. The field of visual programming languages (Chang 1990) has spawned many attempts at visual modelling languages that allow simple, graphics-based construction of systems. Generally, these techniques focus on visual formalisms for specification of *algorithms* (e.g. as an extension of data flow diagrams or flowcharts). One example is the Visual Software Requirements Definition Method (VSRDM) reported by Ohnishi (1994). This method allows the use of arbitrary icons, chosen by the modeller, to construct and animate ("execute") simple models of business processes. The business domain may be decomposed into 'nouns' representing humans, functions, files, data, control and devices. The nouns can interact in complex ways using various pre-specified algorithmic primitives. The method is in some ways reminiscent of the many commercial business process modelling and business process reengineering tools now available.

Although visual programming languages and conceptual modelling techniques like method 'X' are superficially similar (in the use of icons, for example) they have some key differences. Visual programming languages are not (as a rule) based on any explicit psychological principles. But perhaps the most fundamental point of divergence is that conceptual modelling techniques are aimed primarily at representing *business concepts* (as defined in Chapter 1). In contrast, visual programming languages are generally aimed at representing *software designs* and, where they address software requirements, consider them to be an abbreviated or high-level form of software design. This is seen in Ohnishi's VSRDM, for example, where the conceptual noun primitives (*human, function, file, data, control* and *device*) are for the most part computer-related concepts (even *human* corresponds to *end user*, which is a concept defined with reference to computer technology). The business process is modelled as if it were mechanistic and capable of direct automation. Overall, an 'information processing' perspective is used, similar to that found in dataflow diagrams. While this is no doubt a useful approach, it is a very different thing from attempting to capture the business users' *own* concepts, without applying some systems-related perspective(s). The arguments in Section 3.3 about distinguishing technological and business perspectives apply here.

Syntactic sugaring

The concept of syntactic sugar refers to features that are added to a programming language for the sole purpose of making programming easier. The added features are typically redundant in that they provide no new functionality. However, they make the language more palatable. An example of syntactic sugaring is the addition of redundant keywords to programming language constructs in order to make programs more readable. Although syntactic sugar is typically introduced in an effort to help, it also has potential disadvantages. For example, adding new notations that are equivalent to existing structures increases the overall size of the language and creates multiple ways of achieving the same end. Use of the added constructs may obscure the 'actual' mechanism involved, misleading the inexperienced programmer.

Extending the idea to conceptual modelling techniques, we could think of syntactic sugar as otherwise redundant modelling notation added to a conceptual modelling language to make it easier for people to create and to understand models. It could be argued that the use in method 'X' of images (to represent concepts) is a form of syntactic sugaring, since it is intended to make models more understandable. On the face of it, there may be a risk that people will misinterpret method 'X' models because of this; they could interpret the images themselves rather than the structural elements of models. This perceived risk would apply primarily to inexperienced modellers and business users, since we can expect experienced modellers to be well aware of how to interpret their models correctly.

The answer to this question is that we welcome any additional interpretation made possible by the use of images to represent business concepts. For business end users untrained in conceptual modelling, it is the images themselves that help create much of the meaning in a model. Unlike programming languages, conceptual models have a dual role. They simultaneously represent both the business world and software structures (see discussion on conceptual models in Section 1.3). This is particularly true of method 'X'. Its models are structured in a way that is intended to be close to the mental model of a business user, reflecting the business world as perceived by the user. But they are also intended to be used as a step in the design of program and database structures, and hence they represent the structure of the target programs and databases. Because of this dual role we must be careful about what we mean by 'misinterpreting' models. There are two types of misinterpretation that could occur.

Firstly, we might mean that untrained business users will not understand the consequences of their models (i.e. they will be unable to predict the structure of the target programs and databases). This type of misunderstanding is to some extent inevitable. The necessary information can be gleaned only from the structure of a model (i.e. the non-image portions) and still requires certain deductions to be made, much as looking at an object model requires a number of mental leaps to be made before one can visualise the resulting software system. Method 'X' is designed in response to precisely this problem. It is predicated on the prototyping principle: the idea that a concrete demonstration of the consequences of a model (i.e. application structure) is better than having to deduce or predict the likely outcome, at least for inexperienced modellers and untrained business users. Since method 'X' carries over images from the model to the generated application, the images serve as useful visual reminder of the connection between concepts as expressed in the model and the resulting user interface elements in the application.

Alternatively, we might consider the other type of misinterpretation: the risk that people looking at a model will fail to interpret its *business meaning* correctly. A method 'X' model is, after all, intended to be an understandable representation of business concepts. This begs a fundamental question about conceptual models generally, and method 'X' models specifically: *where is the meaning?* Earlier views of conceptual modelling held that models somehow encapsulate meaning ("semantics") and that the role of the modeller is to capture the meaning from the business world and place it "in" a model. In method 'X', a slightly different interpretation of the modeller's role is taken. Method 'X' acknowledges that meaning exists *nowhere except in the mind.* When we look at conceptual models, they remind us of prior experience. Being reminded, we construct meaning (in the form of mental models). The conceptual model itself contains no meaning, only symbols that help us to construct our own. The role of the modeller is therefore to formulate a model (or help others to formulate a model) that allows people to construct or check their own mental models.

Underlying this issue is a question of ownership. Software designers may feel that they own the conceptual models they have a hand in producing. Consequently, the designers may believe that only they can interpret models correctly. But in reality, it is business users who own conceptual models, not software designers, and the business meaning of a model can be found only in the mind of a business user. Obviously, we may hope that two people working with the same conceptual model will tend to construct the same or similar meanings when reminded of business concepts by the model. This is why method 'X' allows images to be chosen by the business users themselves (whether working individually or in a group). The images have meaning to the business users. When they interpret a model using the images, they are not getting the 'wrong' interpretation. In fact, they are getting the *only* correct business interpretation: their own. Other modelling techniques (such as object modelling or entity-relationship modelling) offer business users no chance at all of constructing their own meanings, since they fail to offer understandable representations (see Chapter 3), and are therefore meaningful only to expert modellers. The question is not so much whether misinterpretation will occur, but whether any understanding will be present at all—and, very often, it is not (Bansler and Bødker 1993).

To summarise, we must consider separately the questions of predicting application structure and interpreting business meaning. For expert modellers, we do not need to worry too much about misinterpretation of expected application structure. By definition, experts are people who can be trusted to interpret models correctly in this regard. For inexperienced modellers (e.g. untrained business users) we acknowledge the risk of failure to predict application structure, and respond by providing application preview and generation facilities, so that the consequences of a model can immediately be seen. As to interpreting the business meaning of the model, images that business users have chosen help them to understand the model by reminding them of their own concepts, from which the model's meaning is constructed. The business meaning of a model resides in the mind of the business user. The use of 'syntactic sugar', in the form of images, actually provides better access to meaning than would otherwise be possible for these individuals. The risk of misinterpretation is perhaps greatest for the expert modeller who is not a business domain expert. This person has no suitable prior experience (i.e. relevant business concepts in long-term memory) and is therefore unable to construct an appropriate meaning when viewing a model.

Model analysis

One of the chief benefits in using a model-based approach to software development is the ability to cross-check and otherwise analyse the contents of models. The need for software quality demands that errors and omissions be rectified before expense and effort are wasted on the development of potentially flawed systems. It is often claimed that the cost of fixing software increases by orders of magnitude as one proceeds through the development lifecycle (deMarco and Lister 1987). Therefore it makes sense to find and resolve problems early, at the modelling stage or before.

Several of the best-known and most popular systems development methods incorporate built-in redundancy, the presence of which creates the possibility of cross-checking. One example, OMT (Rumbaugh et al 1991) includes several distinct modelling techniques: data flow diagrams, state-transition diagrams and object class diagrams, amongst others. If all three types of diagram are prepared for a given application then the facts expressed in one diagram can be checked against facts from another diagram. For example, it is relatively easy to compare data flows and data stores (from data flow diagrams) with object attributes (from class diagrams). There are myriad types of cross-checking that can be done. The CASE repository concept (Guinan, Cooprider and Sawyer 1997) grew from the idea that the various diagrams and other representations used in systems analysis and design could share a common information base, which would facilitate the kind of cross-checking mentioned above.

Table 3.12 compares two models of applications software development. The two are extremes, but they serve to highlight a useful distinction. The first (1) describes today's situation. The economics of software development have historically made it necessary to conduct exhaustive analysis and design using multiple *perspectives* (Olle et al 1991) and careful cross-checking between the perspectives. An emphasis has been placed on getting requirements right and spotting design flaws before systems are implemented. Any other path would be risky given the high cost (and high skill requirement) of software development, which is by any standards apt to be slow and labour intensive. When object-oriented languages became popular in the early 1990s it was initially thought that they required less exhaustive analysis and design than non-object-oriented languages. But through experience many organisations learned that careful analysis and design are just as important for object-oriented systems, if not more important. Object-oriented languages do not offer sufficient improvement over conventional third generation languages in speed, cost and skill requirements to warrant a reduction in the emphasis on analysis and design (in fact, there is evidence that they increase the required skill levels and slow down development, for 'one-off' systems at least).

If software development is:	Then there is a need for:
1. Slow Expensive Done by experts	Redundancy – models based on more than one perspective (Olle et al 1991) allowing cross-checking between models. Well-documented and distinct analysis, design and development stages with signoff of intermediate analysis and design deliverables.
2. Fast Cheap Done by anyone	Economy of representation (single perspective) with absence of redundancy. Merged analysis, design and development stages without intermediate deliverables. Checking for analysis and design errors by checking the finished product rather than checking intermediate deliverables.

Table 3.12 Alternative views of software development

The second model (2) in Table 3.12 describes an alternative future situation in which applications software development has become significantly cheaper and easier. Prototyping and the tools used to create prototypes rapidly, such as Visual Basic, have shown how an exhaustive approach to software requirements analysis is not always the best approach. Fourth-generation languages and rapid application development (RAD) tools like Visual Basic have created an industry trend towards prototyping and away from exhaustive analysis and design. Tools that continue this trend might eventually make it feasible to neglect formal analysis and design altogether, or at least significantly reduce their importance. If systems development becomes cheap and quick, and anyone can do it, then there is little justification for employing expensive and highly skilled analysts and designers. Business end users understand their own businesses well enough—better, in most cases, than the average IT professional. Given modelling and prototype generation tools that guarantee reasonably good architecture and usable design, business users could be expected to model business concepts for themselves reasonably successfully. To determine if a model is correct or not, they would simply have to check the resulting applications. There would be no need to agonise over intermediate design deliverables such as conceptual models.

This is the basis for method 'X'. Although restricted in scope, it illustrates how the design process can be automated to produce adequate and serviceable applications. End users merely need to state the relevant business meaning. The acid test they can apply to their models is to inspect and use the resulting applications. There is no need for cross-checking between model elements, or between models, and hence no need for redundancy in model content. In fact, redundancy is to be avoided because of the overhead it creates by duplicating work, and because of the potential for confusion caused by inconsistencies. Exhaustive model analysis is replaced by "80-20" rule and the *fitness-for-purpose principle* (Stapleton 1997).

In this research, application generation capabilities of method 'X' are not used. Instead, we attempt a like-for-like comparison with object modelling, on its own terms. Therefore the option of viewing generated applications was not available and modellers must use other means to test correctness. Method 'X' tool users are offered two ways, apart from viewing generated applications, of checking whether a model is fit for purpose. The first is the *English-language interpretation*. On request, the tool will render all or part of a model in natural language, permitting the user to judge more easily whether the formal meaning of the model is what he or she intends (see Table 3.8). It is often useful to read a model in this way. The natural language version is a formal and rigorous interpretation of the model and as such is sufficiently different from the more usual visual (intuitive, loose) interpretation that it forces one to think in a new way. The second feature is the *'check model'* function, which delivers a quick verdict on the completeness of the model. This helps as an *aide memoir* to remind the modeller of what needs to be done. In 'normal' use of method 'X' neither of these features has any impact on the ability to generate applications; any model, no matter how incorrect or incomplete, can be used to produce an application. As soon as the user views the concrete application, they can begin to see what is wrong with their, more abstract, model.

3.6 Conclusion

This chapter has contributed a number of principles of conceptual modelling (Table 3.13) and a modelling technique is presented that conforms to the principles. The technique will be tested experimentally alongside a conventional object modelling technique (Chapters 4 and 5). The technique is a collaborative one designed to play to the strengths of both groups and individuals. Perception and memory are supported in several ways. It is compatible with the connectionist view of brain function and current ideas about mental models. Unlike many conceptual modelling techniques, it is designed primarily with ease of use and understandability in mind. The technique is oriented towards business end users rather than IT professionals, and it envisages the analyst primarily in the role of facilitator. Nonetheless, model concepts map relatively easily to those of more traditional analysis and design techniques, and the technique is compatible with the design of business applications using contemporary technology.

Representation of models	
1.1	Choose understandable and unambiguous representations (but retain formality if necessary).
1.2	Use conceptual models that match mental models.
1.3	Communicate effectively using the users' own language-in a structured, organised way.
1.4	Maximise bandwidth using visuals where appropriate.
1.5	Maximise comprehension of model concepts by combining words with simplified pictures.
1.6	Use layout effectively and consistently to increase understanding and improve recall.
1.7	Assist recall and comprehension using context and other cues.
1.8	Use a rich set of recognisable 'lifelike' symbols-neither too general nor too specific.
1.9	Consistently use each symbol with one and only one business-related meaning.
Content of models	
2.1	Model concrete concepts, not abstract concepts.
2.2	Model business concepts, not technological concepts.
2.3	Introduce only concepts that exist in the user's world.
2.4	Allow models to be constructed from a rich, extensible set of concepts including unstructured information.
2.5	Increase personal relevance by allowing users to choose their own symbols and by supporting alternative views.
2.6	Support short-term memory using chunking methods (summarising and splitting).
2.7	Allow alternative concept definition methods and context-dependent classification (fuzzy categories).
2.8	Model using concepts at an 'everyday' level of generality.
2.9	Simplify modelling by reducing the number of modelling techniques and removing artificial or fine distinctions in notation.
2.10	Store models such that memory can be supported by semantic associative retrieval.
Modelling process	
3.1	Allow remote modelling at a distance and over time, by virtual groups.
3.2	Use visual aids to support group exploration, learning, negotiation and refinement.
3.3	Provide strong and clear model structure and frameworks to aid group comprehension and thinking.
3.4	Allow groups to work with model components already created and verified by other modellers.
3.5	Support brainstorming by facilitating capture of informal or unstructured ideas and allowing easy model restructuring.
3.6	Tolerate, and reduce the likelihood of, inconsistency and other simple modelling errors.
3.7	Allow alternatives to be compared and explored before decisions are made.
3.8	Allow models to be constructed jointly by smaller groups and individuals.
3.9	Use all means available to retain attention and maintain arousal.
3.10	Give users control and reduce unhelpful sets by avoiding jargon and making both the purpose and method of modelling intuitively obvious.

Table 3.13 Summary of principles

4
Experimental Design

Chapter 3 presented method 'X', a conceptual modelling technique that conforms to specific psychological principles. This chapter describes the design of experiments carried out to test the results when method 'X' was used in practical situations. The overall aim was to determine the relative usability of the modelling technique compared to other techniques, taking into account variables such as the modeller's prior level of experience. To obtain a credible result, careful attention was paid to experimental process and due rigour. Since there is no established way of measuring the usability of a conceptual modelling technique, we chose in this case to compare in a qualitative way against a benchmark technique that could act as a reference point. Object modelling and similar data modelling techniques are well-known and represent current best practice in conceptual modelling, having for some years been used in mainstream systems development methods. The chosen standard was therefore a simple form of object modelling (method 'Y'), as outlined in Chapter 2. Methods 'X' and 'Y' allow like-for-like comparison since they cover similar ground and are intended for similar purposes. The results of the experiment itself are described in Chapter 5, and the results are interpreted in Chapter 6.

4.1 Structure of experiment

Models

The primary experiment consisted of nineteen separate modelling exercises (Table 4.1) carried out by ten modellers (the modellers and their respective organisations are listed in Tables 4.3 and 4.5). Each model was developed using either method 'X' or the control method, object modelling (method 'Y'). Most modellers produced only one model, but eight out of the nineteen models were developed by a single modeller for one organisation (this particular study is outlined below in the section 'Organisations involved'). In total the models consisted of over one hundred separate model versions. After analysis it was found that the inexperienced modellers tended to produce significantly smaller models, on average, using both techniques. This is probably a consequence of the more limited nature of the modelling exercises they carried out. However, a fair comparison between the modelling techniques was still possible since the average model size was roughly consistent between the modelling techniques for each type of modeller. Table 4.2 gives the average model sizes, defined as the total number of concepts expressed in the model (e.g. classes and attributes), ignoring annotations.

Model number and name	Meth-od	Organ-isation	Mod-eller	Group size	Subject matter (as initially agreed)
1. College administration	X	1	E	1	Students, courses, instructors, enquiries, enrolments, fees.
2. Consulting	Y	7	C	1	Consultants, customers, projects, work done, billing rates.
3. Distribution warehousing	Y	6	D	4	Products, warehouses, locations of products in warehouse.
4. Fraud Forum	X	8	J	5	Mobile phone networks, security risks, legislation, counter measures.
5. Fund management	X	5	I	2[1]	Investors, investments, funds, trades, portfolios, banks, brokers, dividends.
6. Help desk	X	2	H	1	Users, problems, technicians, skills, tracking problems, escalation.
7. Homeopathic remedies	X	4	G	1	Remedies, symptoms and indications, diagnosis.
8. Human resources	Y	10	B	1	Staff, skills, sickness, benefits, training, recruitment, appraisals.
9. International roaming	X	8	J	2	Digital services, networks, operators, subscribers, coverage, testing.
10. Legal	X	8	J	4	Data-related legislation, confidentiality, IP, copyright, protection, trade marks.
11. Legal & regulatory	Y	8	J	5	Regulatory and legal issues, meetings, documentation, agendas, actions, advice.
12. Mobile phone billing	Y	8	J	4	Charging principles, service types, tariffs, roaming services, service impact.
13. Operators & networks	Y	8	J	5	Digital mobile phone networks, network operators, membership, services, service areas.
14. Purchase orders	Y	6	D	5	Products, principals, purchase orders, invoices, payments, deliveries, stock.
15. Secretariat	Y	8	J	4	Members, permanent reference documents, working groups, meetings.
16. Security group	X	8	J	5	Audits, standards, algorithms, threats, attacks, certification.
17. Security/ fraud	Y	8	J	3	Counter measures, protocols, security entities, security procedures, products.
18. Stock control	Y	3	A	1	Stock, treatments, sales, orders, shipments, prices, products used.
19. Theatrical productions	X	9	F	1	Students, productions, venues, sfx, lighting plans, cues, performances.

Table 4.1 Models

[1] For model 5, the modeller's main contact was available only sporadically and so the group size was effectively 1 for much of the time.

Experience level	Method	Average model size
High	Method 'Y'	70.2
	Method 'X'	81.7
Low	Method 'Y'	46.2
	Method 'X'	44.8

Table 4.2 Average model size by modeller's experience level and method

The work involved in developing the models took approximately eighteen months, and required negotiation and logistical planning with potential subject organisations and modellers which took place over a two year period prior to that. The effort involved in coding and analysing the resulting model versions was also considerable. Overall, data analysis occupied a period of approximately fourteen months. A secondary experiment was also carried out for triangulation purposes (described later in this section).

Procedure

Each model was produced either in group modelling sessions or by an individual modeller working alone. Group sessions (models 3-4 and 9-17) were run along 'JAD' lines (Wood and Silver 1995) according to common industry practice. The models produced during these sessions were based on the knowledge of group members, who were chosen according to the relevance of their business knowledge and experience. Non-group sessions (models 1-2, 5-8 and 18-19) were produced by individual modellers using their own knowledge of the relevant business areas. Modellers used the method 'X' tool or, for method 'Y', pencil and paper or whiteboard to produce models. In one case (model 5) the lone modeller did not have relevant business knowledge and consulted extensively with a knowledgeable user in interviews.

At the initial stages of modelling each modeller started with a 'blank sheet' and attempted to construct a model from scratch. In subsequent sessions the models were reviewed and revised until the modeller considered them to be complete and correct. This process resulted for each business area in a series of model versions converging on a final version. Each version was assigned a version number and dated. Administrative details were also recorded for each group session including the subject area, session number, time, place and names of participants.

Software tools for modelling were used for both methods, but in a slightly different way for each. Method 'Y' (object modelling) is typically practised without the aid of supporting tools. Instead, white boards, cards or magnetic shapes are often employed and, if a tool is used at all, then this is generally outside of modelling sessions. Conversely, method 'X' is designed to be used with appropriate modelling tools and could not be attempted without them. In this study method 'Y' was therefore supported (for group sessions) in traditional fashion, by whiteboard and overhead projector, and models were rendered using the System Architect CASE tool outside of the modelling sessions (Popkin 1995). Use of method 'X' was supported in modelling sessions by the software tool described in Chapter 3. In the cases of both group and individual modelling sessions, work continued on each model until the modeller was satisfied that a satisfactory model had been produced. For some group models this required additional sessions to be scheduled and the modeller generally spent some time after the last session making final amendments to the model.

The modellers who took part in the study ranged from highly expert data analysts with many years' experience in the IT industry, to business people who had little prior experience with computers. The type and background of both modellers and group members was matched between the two methods as accurately as possible (see next section). In addition, the circumstances of group modelling sessions were carefully arranged so that conditions were as similar as possible to the standard JAD approach, for consistency.

The goal of the primary experiment was to observe modellers in action and to use the models they produced as a data source for 'data mining', to analyse their performance as they created and developed models using the two techniques. The quantitative results were interpreted in the context of the qualitative information gathered during and after modelling sessions, including participant observations, information supplied by participants in questionnaire forms, and interview notes from debriefing sessions with modellers. The outcome of the experiment would have been affected if modellers or group members using one method were exposed to both, since knowledge about one method could easily interfere with understanding of the other. In the case of modeller J this was unavoidable and must be taken into account in the analysis of findings (see Chapter 6). The possibility of other modellers or group members transferring knowledge between modelling techniques was avoided by exposing each member to only one of the techniques. In addition, participants were screened in advance for prior knowledge of object or data modelling techniques. Reliability could also have been affected by factors such as the number of participants in each modelling session, the choice of participants, and the time spent on each modelling session. To allow like-for-like comparison, every effort was made to keep these factors constant between the two modelling techniques. A discussion of the potential effect of a range of such factors is given in Chapter 6.

Modellers

Modellers were selected for participation based on their availability and on their level of prior knowledge, both of modelling and of the relevant business areas. Before starting modelling, each modeller took part in a standard 60-minute practical introduction to the relevant modelling technique. This was followed by periodic reviews to assess progress and assist where possible. The business scope of each model was agreed with the modeller prior to commencement of modelling. To avoid uncertainty about business relevance, each modeller was told that the purpose of the model was to state the data items which a database would need to hold in order to support the specified business area. Modellers were asked to think about (or get group members to talk about) their work and organisations, and to identify relevant business concepts which would then form the basis of the model.

Modeller		**Extent of IT-related knowledge**	**Method**	**Modelling experience**
A	Company director/lecturer	Many years' business experience. Negligible use or knowledge of computers.	Y	None
B	Ex-personnel assistant in a retail bank	Modest use of PCs (word processing, email, simple spreadsheets only).	Y	None
C	Consultancy company administrator	Regular use of PCs (word processing, email, simple spreadsheets only).	Y	None
D	Trainee systems analyst	Regular use of PCs, limited programming, Computer Science degree. Process modelling but no object/data modelling experience.	Y	Some
E	Ex-administrator of a higher education college	Regular use of PCs (word processing, email, simple spreadsheets). Commissioned bespoke software from third parties.	X	Some
F	Audio-visual technician	Little business computer experience; some limited use of computers for art work.	X	None
G	Homeopathic medical practice administrator	Very limited use of PCs (word processing, email, graphics only).	X	None
H	Project manager	Regular use of PCs (word processing, email). Process modelling but no object/data modelling experience.	X	Some
I	4th-year computer science student	No business experience or knowledge. Some programming and introductory-level object/data modelling.	X	Some
J	Senior IT consultant	Many years experience and training in object/data modelling, JAD and facilitation for corporate clients.	X, Y	High

Table 4.3 Modellers

Questions asked by the modellers about modelling technique were answered fully during the 60-minute introduction and follow-up reviews. However, modellers were not told specifically how to model their own business areas. Since assistance was a major potential source of bias, efforts were made to avoid giving greater assistance to one group over the other.

None of the modellers had prior experience of constructing models using method 'X'. Modeller J had many years' experience in object modelling and data modelling using techniques similar to method 'Y' but, of the other modellers, only modeller I had any prior experience of object or data modelling techniques or system design (and this was minimal). Most modellers had no prior experience of software development at all and a minority had no business experience. Modellers were chosen and distributed between the two methods in an attempt to balance the mix of skills, as shown in Table 4.4.

Modelling experience level of modeller	Method	No. models
High (expertise in business, programming, system design and object/data modelling)	Method 'X'	4
	Method 'Y'	5
Some (some experience of computers but little or no programming, system design or object/data modelling)	Method 'X'	2
	Method 'Y'	3
None (little or no experience of computers; no experience of programming, system design or object/data modelling)	Method 'X'	3
	Method 'Y'	2

Table 4.4 Distribution of models by experience level of modeller

In the majority of cases the modeller was already familiar with the area being modelled, having worked in the relevant business for months or years. The exception to this was model 5 (Fund management) which was produced by a modeller who had no business experience and no direct knowledge of the business area in question (investment banking) before starting. Models for organisations 6 and 8 were produced by modellers who had gained their knowledge of the business because the modelling exercises were part of larger modelling projects that they were also involved in. The organisations and modellers taking part in the experiment represented a potentially significant source of experimental unreliability. Because of the small number of individuals and organisations involved it was impossible to rely on any statistical averaging effect and so we cannot claim any form of statistical reliability. Instead modellers were carefully matched between groups and great care was taken to provide a fair comparison between the techniques (see Figure 2.2). In addition, we chose to use qualitative data gathering approach in which participant observation and interpretation played a key part. Preconceived ideas were 'bracketed' (placed to one side). The rigour and care with which the experiment was set up and the qualitative data was gathered and reported represents one of the main ways of ensuring a representative result.

Organisations

Ten organisations were modelled during the study. The models of organisations 1, 2, 4, 5 and 9 were produced mainly for the purposes of this study. The models of organisations 3, 6, 7, 8 and 10 were produced as part of systems development projects. Arranging for organisations to participate in the study proved quite problematic since it involved some risk for the participants and also required suitable projects with convenient timescales. Several organisations agreed to participate but in the event were unable to do so, for various reasons. This led to delays of up to eighteen months as suitable alternative organisations were sought. In the event, models were produced both as part of official IT work by some organisations (models 2-4, 9-18) and also by individuals from other organisations acting unofficially (models 1, 5-8, 19).

Several of the models were produced for organisation 8, an international association of over two hundred telecommunications companies (digital mobile phone network operators). The models were developed as part of a project intended to produce a corporate data model that would underpin future system developments by the organisation. The project was one of a series of initiatives by the organisation that included a workflow (process) analysis of its primary business processes. The part of the project used for this study took place over a four-month period and involved participation by eighteen end-users from the association staff and member companies. Each modelling group used either method 'X' or method 'Y', but not both. Although the areas modelled by the different groups were initially distinct, a good deal of overlap in subject matter was ultimately observed.

Business domain modelled		**Circumstances of modelling exercise**
1	College	Isolated modelling exercise
2	Computer Manufacturer	Isolated modelling exercise
3	Health Spa	Modelling exercise prior to system development.
4	Homeopathic Medical Practice	Isolated modelling exercise
5	Investment Bank	Isolated modelling exercise
6	Retail Goods Distribution Company	Part of larger modelling exercise (20 models, 3 months) conducted as part of system development initiative.
7	Software Consultancy	Part of system development effort
8	Telecomms Business Association	Part of modelling exercise (4 months, several countries) conducted as part of ongoing system development.
9	University Drama Department	Isolated modelling exercise
10	Retail bank	Isolated modelling exercise

Table 4.5 Organisations modelled during study

Since he was unfamiliar with the business and new to the organisation, modeller J went to some lengths to gather as much information as possible in advance of modelling sessions. The international nature of this organisation meant that modelling sessions had to take place in several countries and involved participants from many different locations. Modeller J travelled for this purpose over a period of months and carried a laptop computer with portable data projector so that the modelling tool could be used with method 'X'.

Process of model correction

Once each model was deemed complete by its modeller, and prior to any detailed analysis, the experimenter carefully inspected the model's final state and discussed it in depth with the modeller. Models that met the criteria given in Table 2.3 (consensus, completeness and correctness) were judged to be correct, provided the modeller's understanding was reasonable (which, in all cases, it was). If the model did not meet the criteria a further, corrected, version was created by minimally adjusting the model in an agreed way. The corrected versions were produced without introducing unnecessary changes to the models since this would have caused bias during the correction process. The changes fell into two categories: amending constructs or associations already present (correctness) or adding missing concepts and associations (completeness). In practice it was rarely found necessary to add new concepts. Attributes were largely ignored unless they obviously corresponded to associations that were wrongly represented.

The validity of many of the numerical measures gathered in the experiment hinged on this process of correction. Hence it was necessary to approach it with great caution. The aim was to make the smallest and least significant set of changes possible to a model such that the corrected version met the specific criteria. After discussing the model with the modeller, the experimenter considered possible strategies for correcting the model and chose the one that involved fewest substantive changes. Occasionally this meant correcting a model more than once to see if a corrected version could be produced with fewer changes. Generally it was easy enough to see what the modeller had intended.

A modeller can represent any given situation in a variety of ways, and expert modellers often try to produce elegant models (e.g. by minimising the number of concepts). Both experienced and inexperienced modellers produced models that were functionally correct but, in some cases, relatively inelegant. It was tempting to correct these perceived deficiencies, especially where relatively minor changes would have resulted in significant improvements. However, in the spirit of fairness no such improvements were attempted. Where a modeller had misused notation knowingly and to good effect, this was not treated as an error. But if the modeller misused the notation unknowingly then this was generally a candidate for correction. Overall, corrections were made only where the modeller had made large and obvious errors. In other cases the modeller was given the benefit of the doubt and the model was not corrected.

This leads us to the question of what theoretical basis there is, if any, for the correction of models. The underlying philosophy is related to the arguments given in Chapter 1 on subjectivity. Correction of models is necessarily a subjective act, and can be carried out only with knowledge of the modeller's intentions, an empathetic understanding of the modeller's own view of the model, and an appreciation of the context within which it has been prepared. Hence it was done after discussion with the author of each model and based on a reasonable level of familiarity with the circumstances surrounding the modelling. Correcting models is analogous to correcting or rewording English sentences. Table 4.6 gives some examples where sentences are amended so that they say what the writer originally intended. The first three examples are relatively straightforward. But the fourth is potentially contentious since the corrected version of the sentence is completely unlike the original in form. The flowery terms ("high-quality learning environments" and "facilitation and enhancement of the ongoing learning process") may in fact be well-defined professional jargon with specific and well-understood meanings. Hence, and however unlikely this may seem, it may be inappropriate to replace them with the simpler phrases ("good schools" and "learn properly"). This can be decided only by judgement and after discussion with the writer.

Sentence	**Problem**	**Minimally corrected sentence**
Being weather-damaged and badly infested with termites, I was able to buy the house at quite a low price.	Confused relationship	Since the house was weather-damaged and badly infested with termites I was able to buy it at quite a low price.
Fred went to his brother's house to get his hat.	Ambiguous statement	(If the hat belongs to Fred's brother) Fred went to his brother's house to get his brother's hat.
I've got a luvverly coconut.	Mistake in cardinality	I've got a luvverly bunch of coconuts.
High-quality learning environments are a necessary precondition for facilitation and enhancement of the ongoing learning process.	Vague and complex sentence	Children need good schools if they are to learn properly

Table 4.6 Examples of sentence correction

A similar kind of approach in correcting conceptual models can be used with reasonable results. The same type of restructuring is necessary and similar negotiation is required. Some examples of specific errors in models that were corrected include (a) associations that had been recorded back-to-front (i.e. with cardinalities transposed); (b) categories that had been enumerated (see Chapter 6 for examples); (c) redundant (e.g. circular) associations; (d) important associations that had been omitted from models; (e) inclusion of 'pointer' (foreign key) values that contradicted associations (see Section 3.4). More examples of errors are given in the observations regarding each model in Chapter 5. Obviously, the process of correcting models is potentially contentious and, if wrongly executed, could threaten the validity of the results. However, we made every effort to ensure that it was done in a rigorous and even-handed way, and therefore we can claim that a fair comparison was achieved. That is not to say that a different experimenter would have produced precisely the same numeric values, but we could expect similar results showing the same kinds of trend as were observed in this experiment. The fact that clear and specific patterns were observed repeatedly in the results (see Chapter 6) certainly suggests that this process helped to capture underlying effects that were not an artefact of the model correction and measurement process.

Analysis of results

For each model a selection of versions was chosen for analysis, representing the main evolutionary stages. In most cases fewer than ten versions were used, depending on availability. The analysis involved a comparison of each model version with the final, corrected version of the same model. A number of measures were compared, for each model, that would help later in assessing usability. An automated analysis tool was developed using Microsoft Access to help in tracking model evolution. The analysis tool allowed a succession of model versions to be stored and compared. Data about each model version, including components and their relationships, was encoded in a normalised relational database structure for this purpose. Because of the heavy processing involved in model analysis, each model was analysed separately and the results placed in a summary-level tables for later cross-comparison. The data for each model could be revisited at any time. The tool calculated statistical measures using SQL queries and produced output in the form of tables and graphs, which are reproduced in Chapters 5 and 6.

A diligent modeller will often be prepared to make small but worthwhile adjustments to a model to improve overall correctness. One shortcoming of the analysis tool was that it was unable to differentiate between trivial changes of this sort and more substantive changes. A human observer can recognise the difference between, for example, a trivial name change and one that changes meaning. However, the tool could not recognise two components as the same even if their names were almost identical and, consequently, it failed to recognise associations between components when one of the components' names had been amended. An informal examination of the models showed that component name changes were rare and occurred with broadly the same frequency in the two modelling techniques. Therefore the results of the analysis were probably not compromised, but one should bear in mind that the tool tends to exaggerate the overall number of modelling errors slightly for both methods.

To facilitate analysis, models were loaded into the analysis tool database in the same form regardless of method. Notational differences between the methods were therefore reconciled, as shown in Table 4.12. None of the changes affected the ability to compare model versions.

Issue	Resolution
Types are not recorded in method 'Y', which means that 'wrong type' errors cannot be reported.	The number of 'wrong type' errors was negligible and so they were excluded from the analysis.
The CASE tool used for method 'Y' models (System Architect) sometimes inserts foreign key attributes (into classes) which it derives from associations. In addition, some models lacked attributes.	None (as a consequence, some models contain slightly more attributes than they would otherwise have, and some contain none).
Supertype/subtype relationships were not used in method 'X' models	Relationships of this type were excluded from the numerical analysis.
Pictures and positional information are not available in method 'Y'	Pictures and positional information used in method 'X' models were excluded from the analysis.
Association names (in method 'Y') and role names (in method 'X') were used intermittently and infrequently.	Association and role names were excluded from the analysis.

Table 4.12 Notational differences reconciled to allow analysis

Secondary study

An additional experiment was carried out to provide triangulation. In the secondary experiment 46 models were analysed more briefly. The models were produced using method 'Y' by modellers who had received brief training in the same method. The experiment was intended to give a rough indication of the level of error one might expect from individuals with some knowledge of modelling technique but little practical experience (that is, inexperienced modellers). The subjects in this experiment were first-year and third-year students enrolled in a B.Sc course in computer science. Both sets of students had received the same brief introduction to object and data modelling of about 3 hours, but neither had practical experience of applying the modelling techniques on non-trivial models. Simple written descriptions of three of the subject areas used in the main study were given to the students to model individually. Each student was allowed adequate time to produce what they judged to be a finished model.

4.2 Measures

It is possible to establish multiple facets of a concept of interest so as to aid in 'operationalising' it for research purposes (Lazarsfeld 1958). To measure concept X we can instead measure concepts X_1, X_2, X_3, and so on, each of which refers to a specific aspect of X. To measure model quality we therefore measure aspects of quality (completeness, correctness, error rate, accuracy, etc.). To calculate values for these facets we needed to measure specific attributes of each model, which are discussed below in the section of quantitative measures. A 'factor analysis' can help discover which facets may be considered together as 'factors' (for an example in an organisational context, see Dunham, Aldag and Brief 1977). Facets are grouped as a single factor if they tend to be strongly correlated. It may be that some of our facets (e.g. completeness and correctness) are strongly correlated and therefore constitute a single factor (which we might refer to as 'model quality'). Factor analysis can be carried out only after measurements have been made. To determine whether the facets we used were indeed representative of a single factor we computed the correlation of each facet with every other facet (see Chapter 5).

Quantitative measures

After a final, corrected version of each model had been produced (for those models that required it), each model version was analysed to determine the raw measures listed in Table 4.7. The process of model analysis also yielded observations by the experimenter of a more qualitative nature, and these have been incorporated in the commentary attached to each model in Chapter 5.

The measurements were obtained using a method that was applied rigorously and without exception. To ensure reliability all values were measured automatically rather than by inspection. Models were put into electronic form (if not already) allowing their contents to be analysed by a tool developed especially for this purpose. This method permitted a very exhaustive form of measurement. It also allowed a form of exploratory analysis that would otherwise have been impossible. For example, the graphs in Chapter 6 showing emergent relationships between model size and change rates were produced only after a good deal of alternative analyses were tried. To ensure validity, samples were analysed by hand and the results compared with automatically-generated analyses. The exercise was repeated until the analysis program had been thoroughly debugged and reliably reproduced the same results as hand analysis. The results were tabulated and also plotted graphically so that a visual assessment could be made. Putting the results into a graphical form also paid dividends since it allowed striking emergent patterns and correlations to be observed, which are detailed in Chapter 6.

Measure	Description
Number of components (c)	The total number of model components in the model version. For method 'Y' this refers to definitions of object classes and attributes. For method 'X' this refers to definition of any non-annotation components. In both cases each definition counted once only and multiple usage of any definition was ignored. Every component was counted regardless of how detailed (or otherwise) its definition was.
Number of finished components (c_f)	The number of components which are identical to those in the corrected model (i.e., with identical attributes, relationships, links, keys, etc.).
Number of attributes (c_a)	The total number of distinct data items defined in the model (excluding reused definitions).
Number of relationships (r)	The number of relationships of all types between model components, excluding annotations.
Number of correct relationships (r_c)	The number of correct relationships (i.e. relationships that are identical to those in the corrected model). This measure excludes relationships that are expressed between the wrong components, relationships named poorly and relationships enumerated incorrectly.
Number of changes (m)	The total number of separately identifiable changes made to the model between one version and the next. Each change was classified for later analysis.
Number of errors (m_e)	The total number of separately identifiable errors of any type present in the model version, including omissions. Each error was classified for later analysis.
Number of corrections (m_c)	The total number of separately identifiable changes to the model, between one version and the next, that corrected errors.
Modelling time (t)	Total time (in minutes) spent on modelling over all versions.

Table 4.7 Measurements taken from models

Calculating effectiveness and usability

The ratios used to evaluate model development are defined precisely in Table 4.8. Primed variables in the table refer to values measured for the corrected version of each model. Using the values listed in Table 4.8 a single measure 'effectiveness' was calculated for each model. This measure represents the overall effectiveness of the modeller and gives a rough indication of a modeller's ability to construct a complete model, without making mistakes, in good time. It is calculated using the formula:

$$E = \frac{q_1 q_2 p(1-e)}{200}$$

where q_1 is the percentage completeness of the final uncorrected version of the model, q_2 is the percentage correctness of the final uncorrected version of the model, p is the modeller's productivity, and e is the modeller's error rate. A 'perfect' error rate is zero (no errors at all) whilst an error rate of one means that every change made to the model by the modeller was an error. The effectiveness value is divided by 200 simply to produce a figure in a range comparable to the other measures used in this study. The calculation does not use the measure 'complexity' since there is no clear relationship between complexity and quality.

Facet	**Description**	**Formula**
Completeness (q_1)	The percentage of model components and relationships in the finished version that are also present in this version.	$100 \times \left(\frac{c_f + r_c}{c' + r'}\right)$
Correctness (q_2)	The percentage of components in the current version that are present and defined correctly (i.e. in the same way as in the finished version).	$100 \times \left(\frac{c_f + r_c}{c + r}\right)$
Attribute ratio (r_a)	The percentage of model components in the current version that are data items.	$\frac{a}{c}$
Volatility (v)	The total number of changes in the current version of the model relative to the total number of model components and relationships.	$\frac{m}{(c+r)}$
Accuracy (a)	The percentage of changes in the current version of the model that are corrections (i.e. changes that cause the model to become closer to its completed form).	$100 \times \left(\frac{e_c}{m}\right)$
Error rate (e)	The average number of errors made for each business concept in the finished (corrected) model version. (Calculated once for each model across all versions).	$\frac{\sum_v (m - m_c)}{c'}$
Productivity (p)	The average number of finished (i.e. correct) business concepts produced per hour of modelling time. (Calculated once for each model across all versions.)	$60 \times \left(\frac{c}{t}\right)$
Complexity (x)	The average number of relationships for each model component.	$100 \times \left(\frac{r}{c}\right)$

Table 4.8 Measures used to evaluate model development

Table 4.9 allows effectiveness scores to be classified as either *very good, good, poor* or *very poor.* Although the classifications are arbitrary, they permit comparison and help to avoid an unwarranted concentration on the specific scores.

Level	**Score**	**Description**
Very good	>150	Correct and complete. Suitable for use in system design without refinement. Produced quickly and with few errors.
Good	101-150	Largely correct and complete, but usable only with further work (or produced relatively slowly).
Poor	51-100	Coherent but substantially incomplete and incorrect. Usable only as a 'first-cut' model.
Very poor	0-50	Incoherent and/or grossly incorrect. Produced very slowly and/or with many errors. Unusable even as a 'first-cut' model.

Table 4.9 Effectiveness levels

The effect of the modeller's ability was then factored out to produce an average usability figure for each modelling technique. In very rough terms a modeller's overall effectiveness is due to his or her own ability, skill and knowledge, together with the inherent usability of the method being employed. The following formula was used:

$$U = E - A$$

where E is the effectiveness of the modeller using a particular method (as computed above) and A is a measure of the modeller's estimated ability level.

Qualitative measures

There were three main sources of qualitative results: questionnaires completed by modelling session participants (modeller and group members), notes from interviews with modellers, and observations by the researcher. The questionnaire forms used during the experiment are reproduced in Appendix D. A video recording was also made of one modelling session conducted by the expert modeller but, owing to technical problems, little useful information could be gleaned from the recording and so no results are presented here. Chapter 2 gives a detailed account of the steps taken to ensure that qualitative data was both valid and reliable and that it was gathered and analysed in a rigorous way. The discussion below summarises the main data-gathering methods employed.

Modellers in group modelling sessions were asked to observe and note the actions and comments of participants. In addition, each participant was asked to complete a series of brief questionnaires. Before the sessions, participants were asked in a questionnaire to supply details of their prior training, background knowledge, experience using similar modelling techniques and any prior participation in similar studies. The aim here was to determine the participants' attitudes and assumptions and their prior knowledge and experience in business, conceptual modelling, workflow analysis, information systems development and other relevant areas. After each workshop session, participants were asked in a second questionnaire about their impressions and understanding of the techniques that were used. They rated numerically the completeness and correctness of the models. The aim was to establish a snapshot of participants' thinking in each session, against which later results could be compared. In particular, the intention was to compare the participants' reported assessments of completeness and correctness with the levels of completeness and correctness as later determined in numerical analysis of models. The purpose of using questionnaires rather than interviews in the above three situations was to help overcome the potential influence (inhibiting or otherwise) of the interviewer's own characteristics, such as age, appearance, race, gender and social class, which can have a significant effect on interviewees' responses (Sudman and Bradburn 1974).

Notes taken by the modeller during group modelling sessions were also intended to act as a qualitative record of the proceedings, and their content depended on whatever seemed important to the modeller at the time. The modeller recorded an assessment of the level and type of input by each participant as well as specific areas of difficulty, confusion or clarity for each person.

Management consent was obtained before the experiments were carried out, but the group members were not told they were participating in a research exercise. The advantage of using covert observation was that effects deriving from the experimental nature of the sessions could be avoided. Both modelling techniques were presented as if they were a natural part of the process. From an ethical point of view, the use of covert experimental techniques can present difficulties. It is hard to justify a situation in which individuals are involved in experiments unwittingly and may suffer some negative consequences as a result. The experimenter must seek and obtain the informed consent of the participants before taking part. The possibility of a negative outcome was minimised in this instance by checking beforehand that viable models could be created using method 'X'. In addition, a more experienced modeller than usual carried out the modelling. The progress of the experiment was closely monitored by the experimenter, who was prepared to step in and halt proceedings if problems arose. The fallback position would have been to revert to the use of method 'Y' exclusively. In the event, no need for this arose. Realistically, the risk to the subject organisation accruing from the use of method 'X' was low and was probably exceeded by the normal level of risk attaching to any conceptual modelling project.

4.4 Summary

This chapter has described an experimental design in which a main experiment and a secondary experiment were carried out. The main experiment consisted of nineteen separate modelling exercises conducted in business organisations by modellers of varying experience levels. The modelling exercises used two conceptual modelling techniques: methods 'X' and 'Y'. Method 'X' is the technique presented in Chapter 3 of this thesis, which conforms to a number of carefully justified psychological principles of conceptual modelling. Method 'Y' is a simplified conceptual modelling technique similar to mainstream object and data modelling techniques, as outlined in Chapter 2. For both techniques, some modelling sessions were conducted by groups and some by lone modellers. Throughout the experimental portion of the research careful attention was paid to due process, validity and reliability. Chapter 2 gives a thorough account of the measures that were taken to ensure experimental rigour with regard to both qualitative and quantitative data. In addition, Chapter 6 presents a comprehensive critique of the experiment and the overall methods employed in the larger research project, assessing the potential impact of various factors that could have influenced the results.

Table 4.10 (overleaf) gives a complete list of all data gathered and derived in the experiment, by source. Each item is classified to indicate the form it takes. Data was gathered by participant observation, questionnaire, interview and by numerically analysing the resulting models to provide measures of model quality and modeller effectiveness. The model analysis technique allowed the development of models to be tracked over time. In the secondary experiment a larger set of inexperienced modellers produce models working alone. Developmental patterns in the secondary experiment were not analysed but overall figures for model quality were produced, which allowed triangulation with the results of the main experiment. The results of both experiments are set out in Chapter 5 and the results are interpreted comprehensively in Chapter 6.

Dimensions	Source	Measure	Coding
Group member	Questionnaire	Prior training (education)	No degree/IT degree/other
		Relevant business expertise	N/A
		Participation in modelling	Yes/No
		Knowledge of modelling	Yes/No
		Use of computers	Every day/sometimes/ rarely/never
		Programmed before?	Yes/No
		Created a database before?	Yes/No
Model	Observation	Total modelling time	Total mins
	Questionnaire	Comments by modeller	N/A
	Derived	Error rate	Ratio
		Productivity	Ratio
		Effectiveness	0-200
Model version	Model analysis	Number of components	Count
		Number of valid components	Count
		Number of finished components	Count
		Number of attributes	Count
		Number of relationships	Count
		Relationship accuracy	Count
		Number of errors	Count
		Number of changes	Count
		Number of corrections	Count
	Derived	Completeness	Percentage
		Correctness	Percentage
		Attribute ratio	Ratio
		Volatility	Ratio
		Accuracy	Percentage
		Complexity	Ratio
Group member and model version	Questionnaire	Purpose of modelling	N/A
		Value of modelling (rating)	Scale 1-5
		Value of modelling (explanation)	N/A
		Understanding of technique	Scale 1-5
		Understanding of software	Scale 1-5
		Completeness of model	Scale 1-5
		Correctness of model	Scale 1-5
		Overall impressions or comments	N/A
	Observation	Type of input	High/medium/low
		Level of input	Scale 1-5
		Specific areas of difficulty	N/A
		Specific areas of clarity	N/A
		Comments by modeller	N/A
		Observations by experimenter	N/A
Modelling technique	Derived	Usability	Calculation

Table 4.10 Summary of measures

5 Findings

This chapter presents the findings from the experimental research described in Chapter 4, consisting of a main experiment with nineteen modelling exercises and a more limited secondary experiment. The results from both parts of the experiment are presented here with as little interpretation as possible. The results include statistics derived from analysis of models, observations by participants, and a summary of the questionnaire responses. Interview notes are given in full in Appendix C. The experimental results are interpreted comprehensively in Chapter 6.

5.1 Analysis of models

The numerical information presented below was derived directly from models using the analysis tool described in Chapter 4. A brief commentary is included for each model pointing out relevant features. The models themselves are reproduced for convenience in Appendix E.

1. College Administration Model

This model was produced by the ex-administrator of a college of higher education. The modeller had no prior experience of computer programming, database design or modelling techniques, but had been involved in commissioning and overseeing the development of software by third parties. Hence she is reasonably familiar with what can be expected of software systems. The modeller was enthusiastic about using method 'X' and clearly found it intriguing, having been used to a much slower process of requirements discovery. The modeller observed that the use of pictures in method 'X' helped make models readily understandable.

In this model the error count initially began to decline (the absence of correct components and relationships is treated as an error, so all models started off with a non-zero error count). But the error count soon started to increase again, largely due to the creation of unwanted relationships and, to a lesser degree, creation of unwanted components, from versions 1 to 8. Eventually the modeller identified and resolved these errors, and over the remaining 14 versions the number of errors declined rapidly. This model exhibited an interesting pattern of development in that there were three main bursts of activity during its creation, at versions 2, 9 and 16. Most of the changes made at versions 2 and 16 were correct, but in version 9 the model became significantly less correct. Discussions with the modeller seem to indicate that this came about because of her 'learning curve'.

The initial simplicity of the model and the modeller's early success in creating components gave her confidence. Identification and addition of components predominated at this early stage. But after version 4, correctness began to decline sharply and, by version 9, the model was both highly complex and highly incorrect. The modeller had realised by this stage that she would have to find ways of structuring the model more correctly. Mistakes were made as she tried to do this, most of which were due to the addition of incorrect relationships. However, the modeller continued to work with the model and, by version 16, it was clear which relationships had to be corrected. The changes recorded at version 16 are equally split between removal of incorrect relationships and addition of new, correct relationships.

The modeller's productivity was very high and the overall error rate intermediate (approximately one error in every four changes). The completeness and correctness levels were high well before the model was completed. Overall, the complexity of the model increased rapidly and reached its final level after only a few versions. The quality of the model was judged to be sufficient that no corrected version was considered necessary. Table 5.1 summarises the main statistics for the model. The figure for "No. of components" refers to the total number of business concepts and data items, excluding annotations.

No. of components	38	Changes per business concept	4.41
No. of relationships	44	Errors per change	0.28
Relationships per business concept	4.40	Business concepts per hour	4.00
Attributes per business concept	2.80	No. of changes	195

Breakdown of changes	Total	%
Relationship correctly removed	44	22.56
Incorrect relationship added	42	21.53
Relationship correctly added	41	21.02
Component correctly added	39	20.00
Incorrect component added	10	5.12
Component correctly removed	10	5.12
Cardinality corrected	4	2.05
Relationship wrongly removed	2	1.02
Component wrongly removed	1	0.51
Component type corrected	1	0.51
Cardinality altered wrongly	1	0.51

Table 5.1 Results for model 1

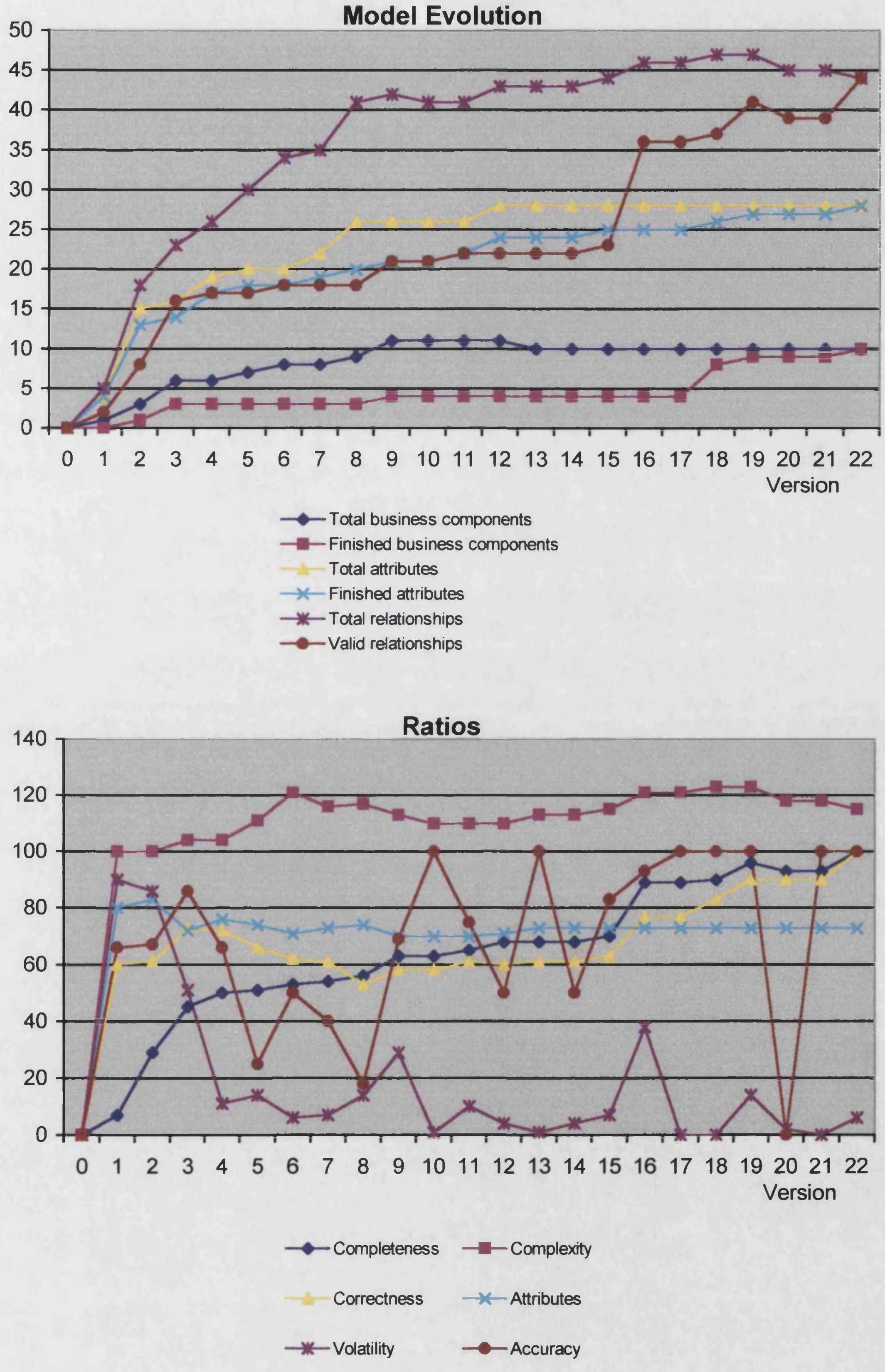
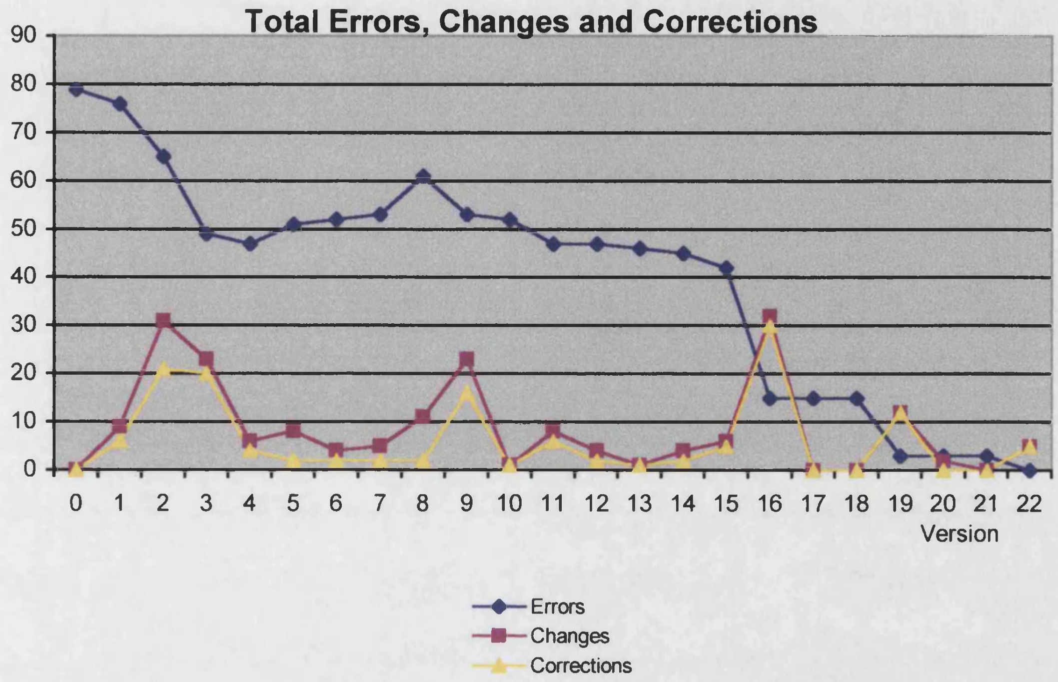


Figure 5.1 Results for model 1 (a)



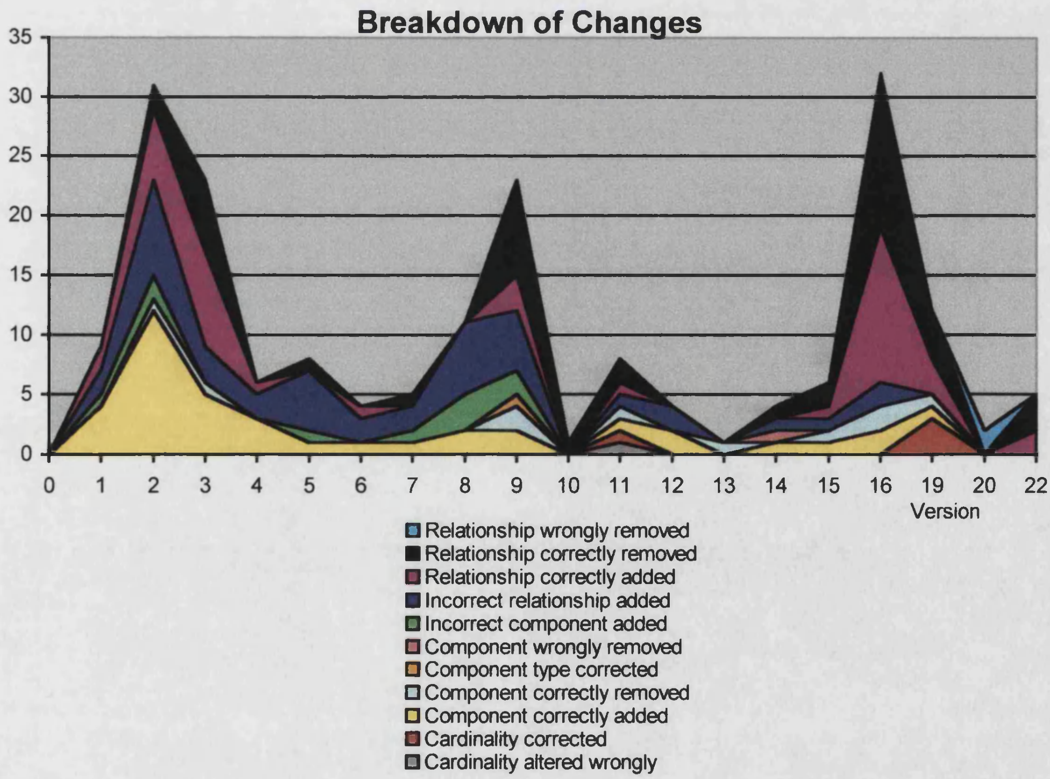


Figure 5.2 Results for model 1 (b)

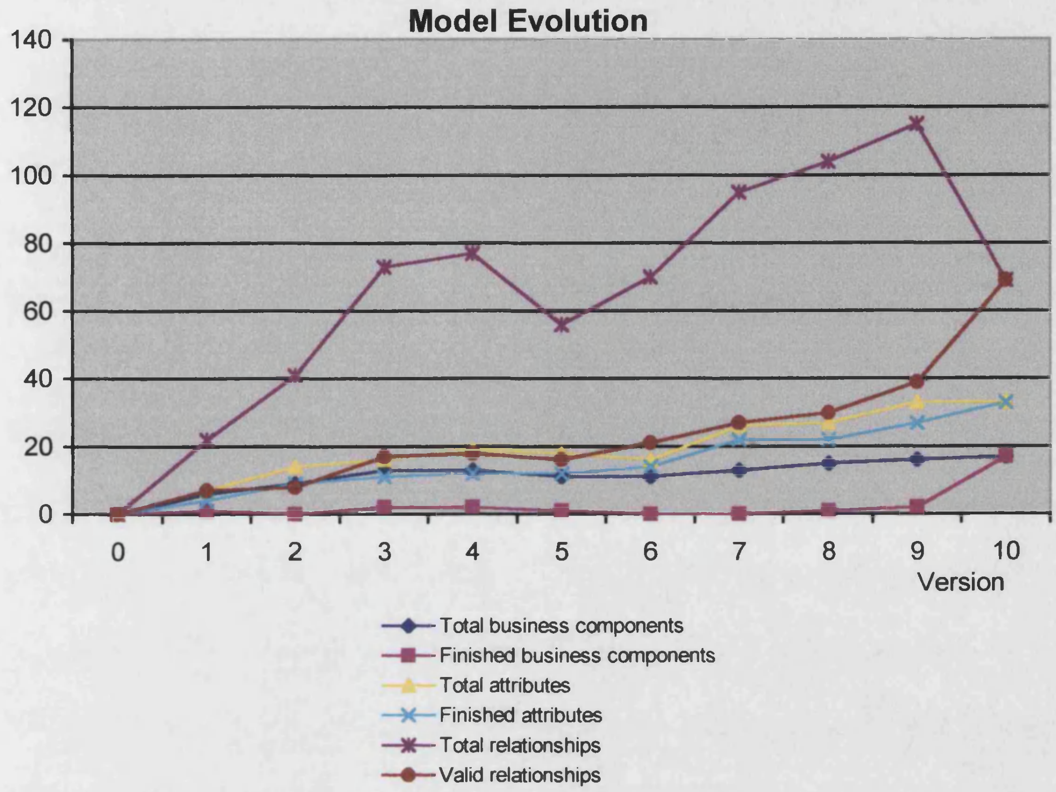
2. Consulting Model

This model was produced by an office manager without experience of system development or modelling. The modeller was enthusiastic and serious about the modelling and took great care with each version, deliberating for some time over the most appropriate structure. She seemed happy with the modelling technique and appeared to understand the basic principles of structuring models. The modeller made many changes to the model, more than half which were mistakes. The model quickly became drastically over-complex with many incorrect relationships. By the time the modeller was satisfied with the model (version 9), only about one-third of all relationships were correct. The majority of relationships were either entirely redundant or related the wrong concepts. Because of this the number of errors in the model climbed early on and, although it eventually fell back slightly, remained high. A corrected version of the model was produced (version 10), which included roughly the same components, but related them using approximately 40% fewer relationships. Overall, productivity was poor.

No of components	50	Changes per component	8.04
No of relationships	69	Errors per change	0.51
Relationships per component	4.05	Components per hour	1.33
Attributes per component	1.94	No. of changes	424

Breakdown of changes	Total	%
Relationship correctly removed	119	28.06
Incorrect relationship added	116	27.35
Relationship correctly added	70	16.50
Component correctly added	54	12.73
Incorrect component added	20	4.71
Component correctly removed	20	4.71
Cardinality altered wrongly	9	2.12
Relationship wrongly removed	6	1.41
Component wrongly removed	4	0.94
Cardinality corrected	4	0.94
Component type corrected	2	0.47

Table 5.2 Results for model 2



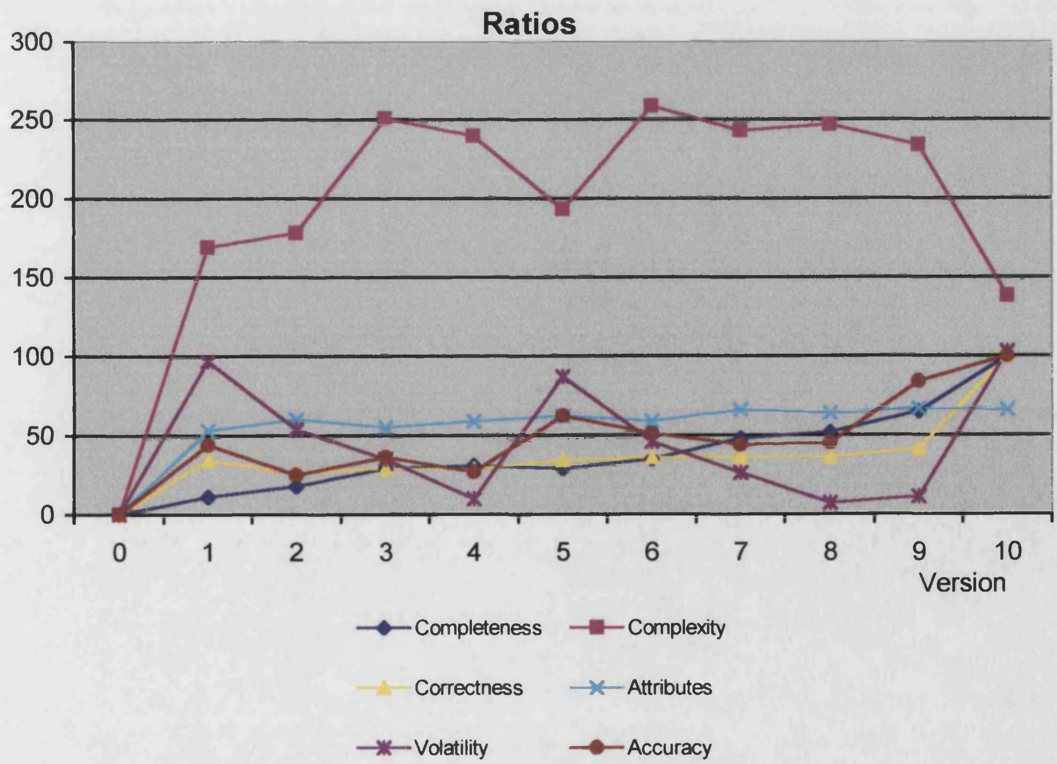
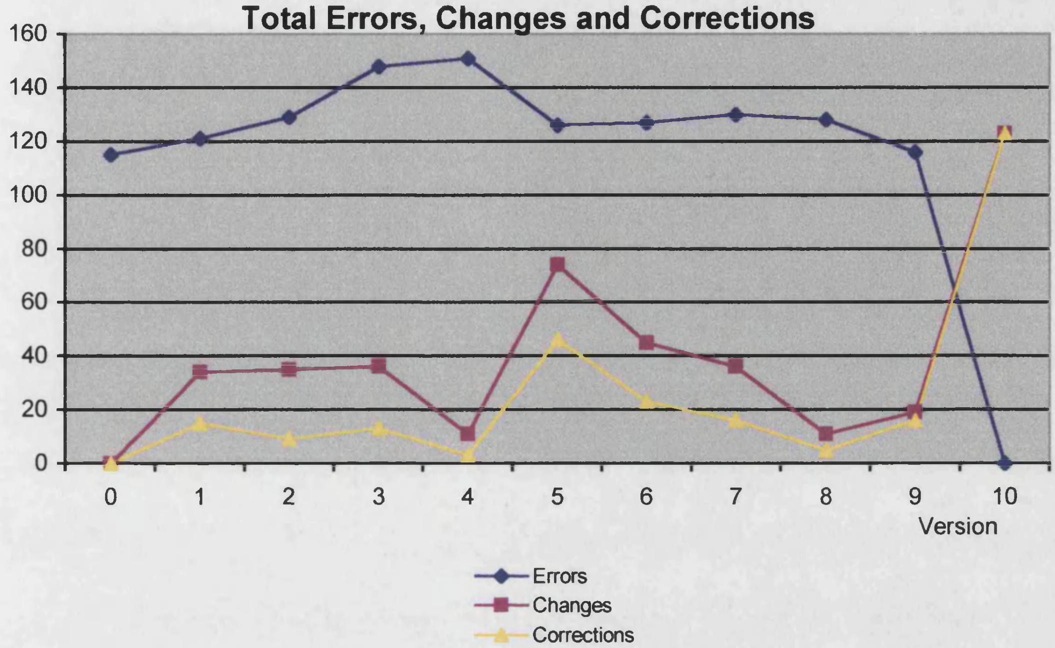


Figure 5.3 Results for model 2 (a)



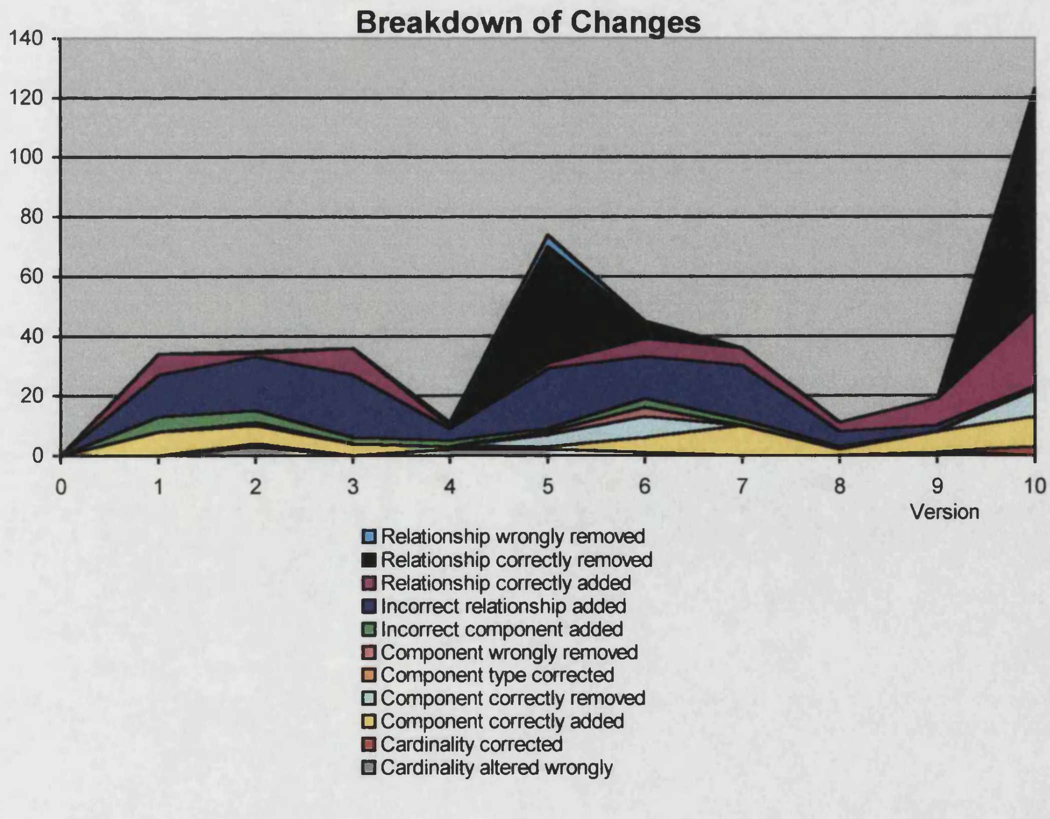


Figure 5.4 Results for model 2 (b)

3. Distribution Warehousing Model

This model was produced by a trainee systems analyst with a degree in computer science and several years' experience in the IT industry, but no prior object or data modelling experience. The modeller had gained about 12 months' experience using requirements analysis and process modelling techniques, and had been coached in the use of method 'Y' prior to and during the production of this model. She was familiar with the subject area, having spent the previous six months working in the same business. The model was initially drawn on a whiteboard and then transcribed onto paper. Details of classes and attributes were keyed into a CASE tool (System Architect) which was then used to create diagrams. The resulting diagrams were photocopied onto overhead projector foils for use in subsequent modelling sessions. Only one uncorrected version of the model was available (version 1) but it is clear from this that the modeller deviated significantly from what was required. The total number of errors in the model (including missing components and relationships) initially increased instead of declining. The modeller made several errors repeatedly:

- Writing relationships backwards (i.e. stating the cardinalities the wrong way around), especially where categories (such as 'Status' or 'Type' classes) were related to other classes.
- Relating classes illogically. For example, 'Preferred (product) location' is modelled as a property of 'Warehouse location' instead of 'Product'.
- Including duplicated concepts. For example, customer address is included as a property of both 'Customer account' and 'Sales order'.
- Omitting important relationships altogether.
- Using an 'intuitive' or loose modelling style. For example, the modeller framed classes vaguely such that they that did not obviously correspond to any identifiable business concepts. One example is 'Customer account' which in the model has one 'Direct credit' and one 'Direct debit'. In reality a customer account would have zero or many transactions and each transaction would be either a credit or a debit ('direct' is a misnomer here).
- Enumerating categories (e.g. 'Warehouse rack' contains attributes 'Heavy' and 'Light' whereas these are in fact possible values of 'Rack type').
- Including redundant, circular relationships. For example, 'Pallet activity' is related to 'Pallet activity type' which is related to 'Pallet activity reason', which is in turn related to 'Pallet activity'.
- Making relationships mandatory when this cannot be honoured (e.g. each customer account must be on a route, even though routes are planned only after customers are acquired).

Overall, the modeller spent a good deal more time making and correcting errors than on anything else. She used a markedly 'impressionistic' style of modelling in which the meaning of the model depended heavily on her own interpretation, which tended not to follow consistent rules. One mistake in interpretation was made consistently: relationships were read as if their scope were restricted by other, nearby relationships. This error was in fact made by *all* of the inexperienced modellers. It amounts to trying to read the 'whole meaning' of the model in each relationship, where in fact the whole meaning of the model is the sum total of all individual concepts and relationships. This confusion in reading relationships also led to incorrect formulation of relationships. The modeller intended the version represented in the graphs as version 1 to be the finished version of the model, and was pleased with her work, being unaware of any problems in the model.

No assessment of productivity was possible since the modelling time was not recorded. However, according to the modeller "a long time" was spent on the model. Despite its small objective size, it was considered a major piece of work by the modeller, who had not produced object models before.

No of components	29	Changes per component	4.81
No of relationships	63	Errors per change	0.59
Relationships per component	2.17	No. of changes	140
Attributes per component	0.00		

Breakdown of changes	Total	%
Relationship correctly removed	31	22.14
Incorrect relationship added	31	22.14
Relationship correctly added	29	20.71
Component correctly added	29	20.71
Incorrect component added	10	7.14
Component correctly removed	10	7.14

Table 5.3 Results for model 3

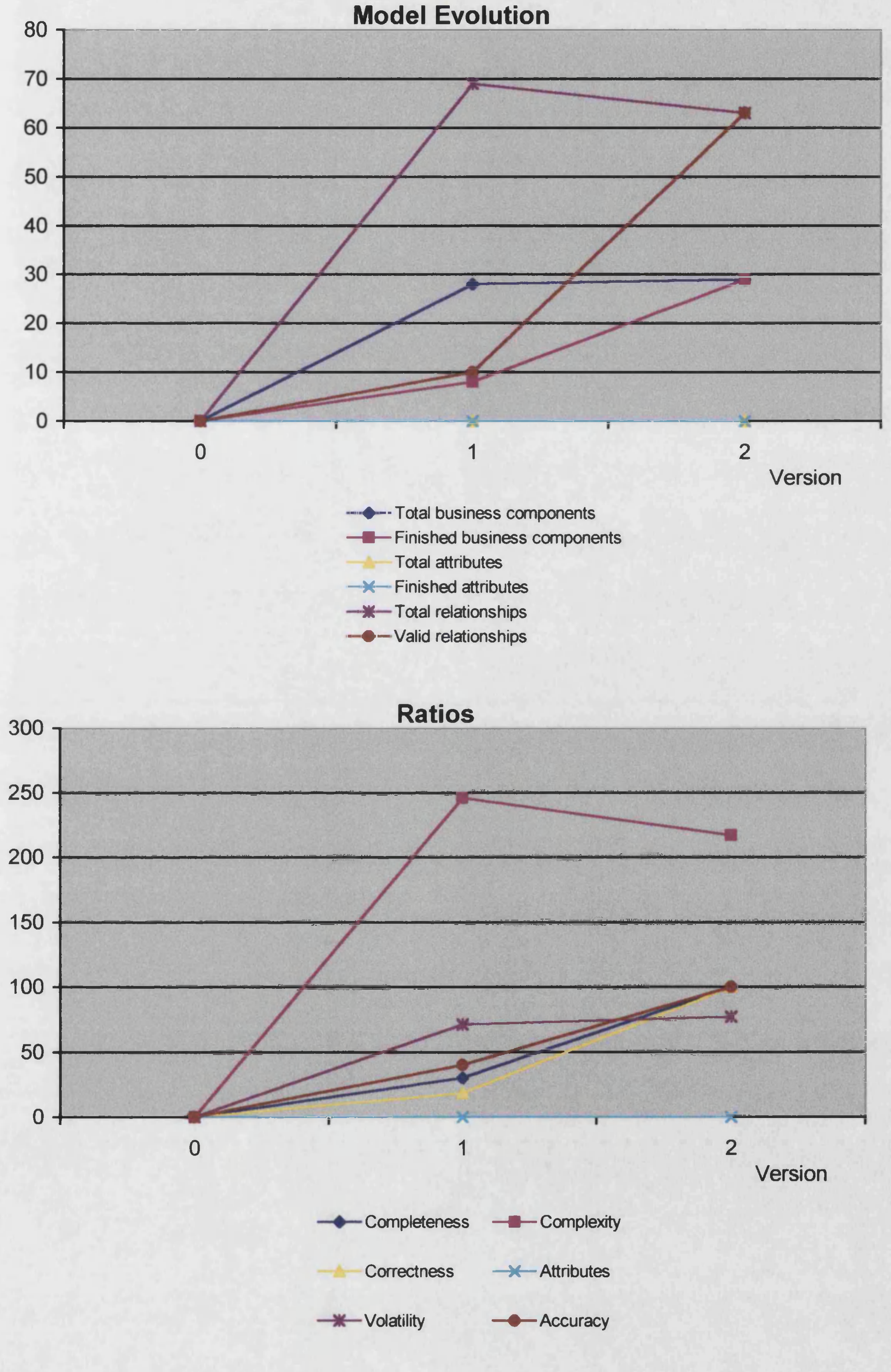
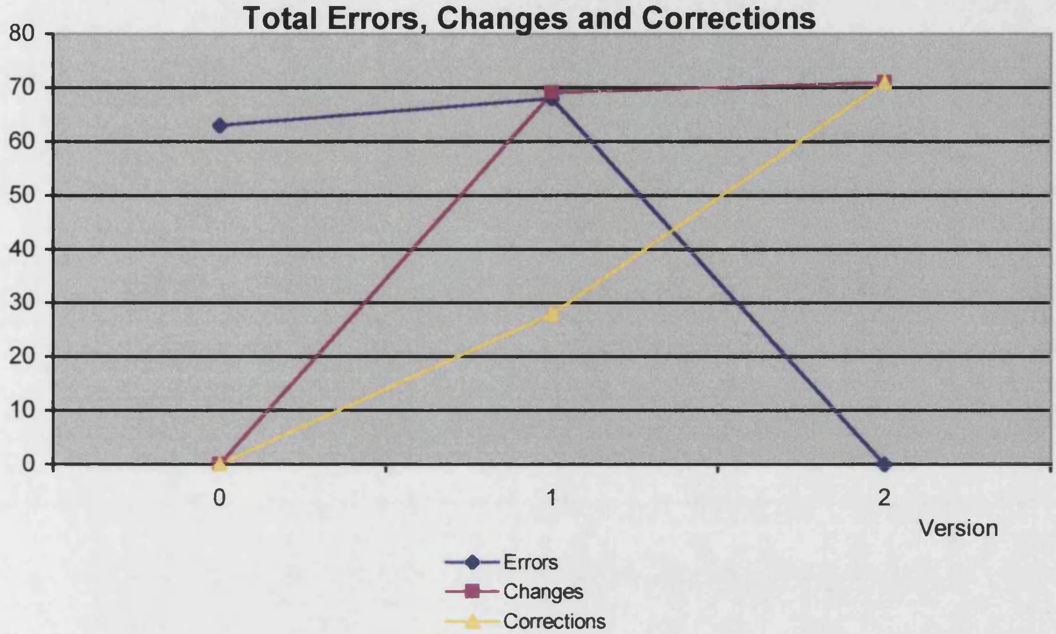


Figure 5.5 Results for model 3 (a)



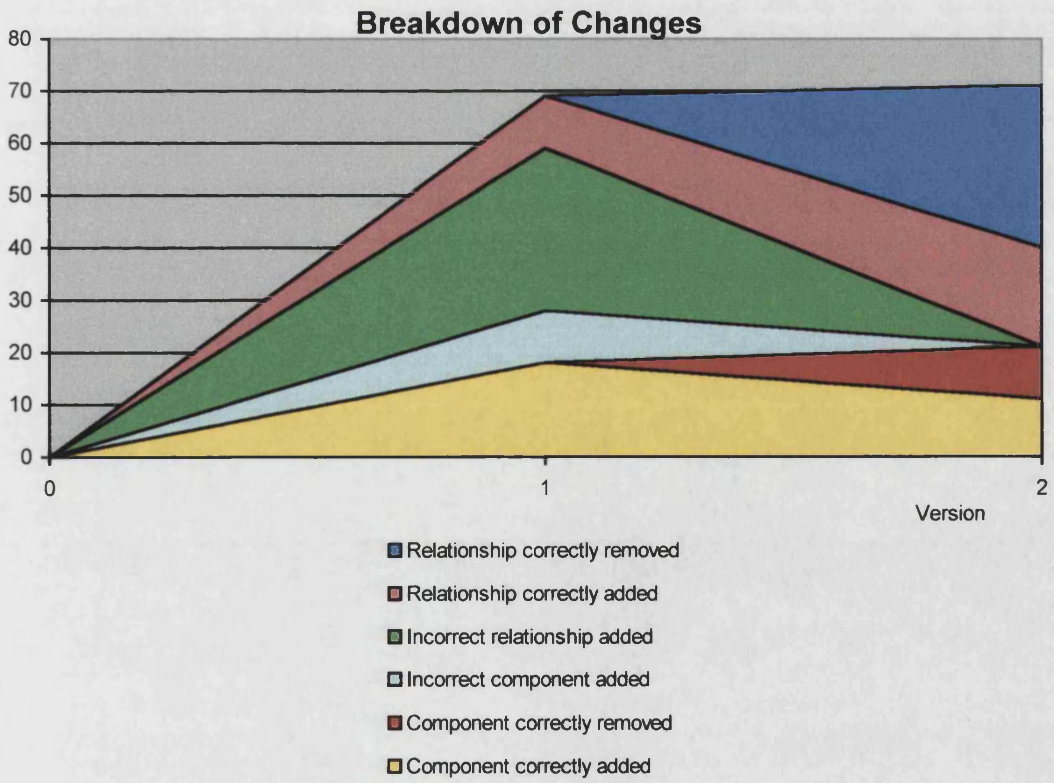


Figure 5.6 Results for model 3 (b)

4. Fraud Forum Model

This small model was produced by an experienced analyst. The model was trivial and barely developed beyond an initial statement of objectives, because the subject matter was incorporated into a different model for operational reasons. The model has been retained for the sake of completeness only, and has been excluded from further analysis in Chapter 5.

No of components	27	Changes per component	1.28
No of relationships	25	Errors per change	0.00
Relationships per component	0.92	Components per hour	1.80
Attributes per component	0.00	No. of changes	53

Breakdown of changes	Total	%
Component correctly added	41	77.35
Relationship correctly added	12	22.64

Table 5.4 Results for model 4

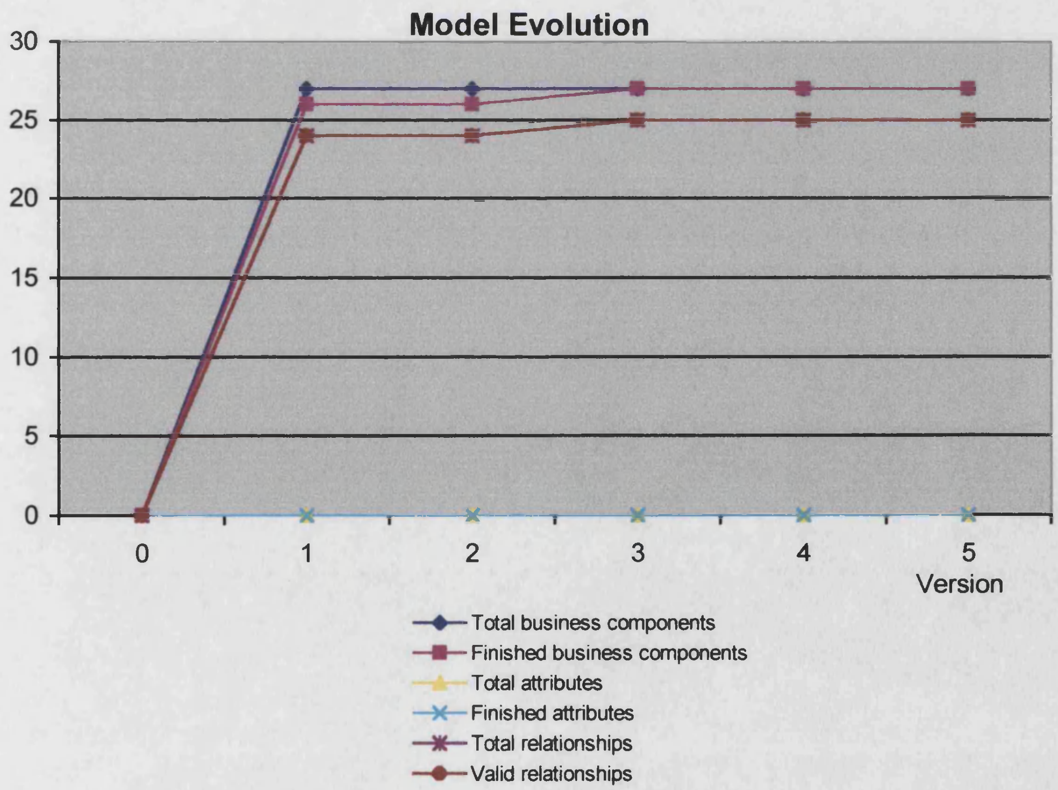
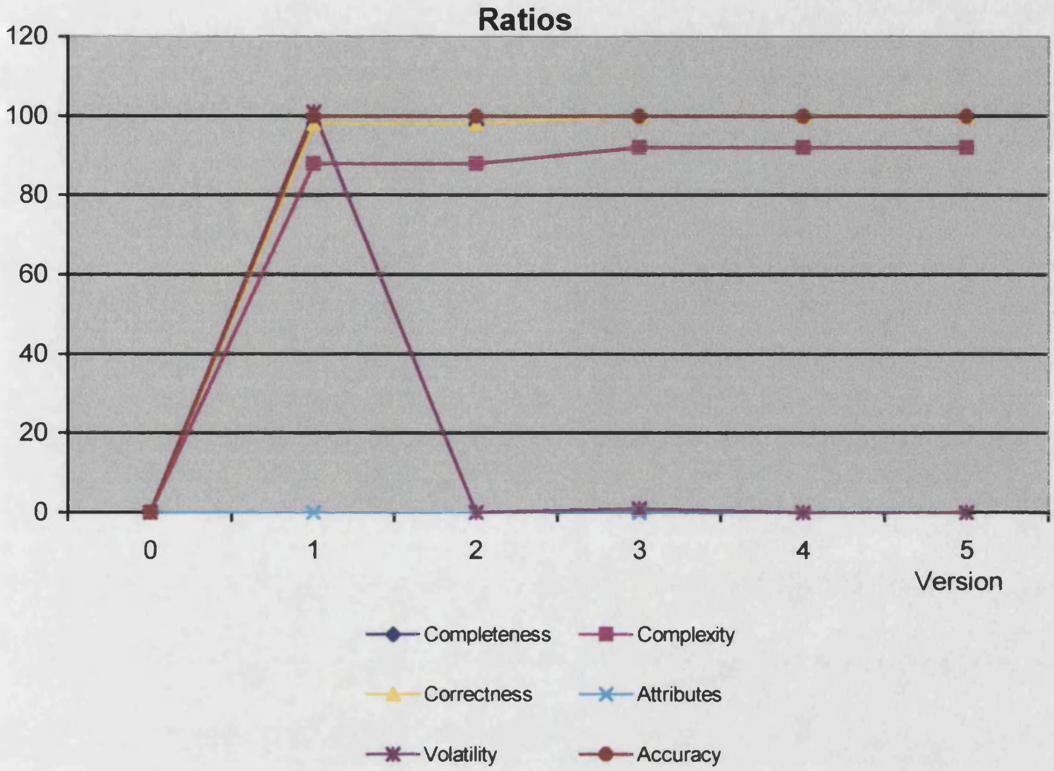


Figure 5.7 Results for model 4 (a)



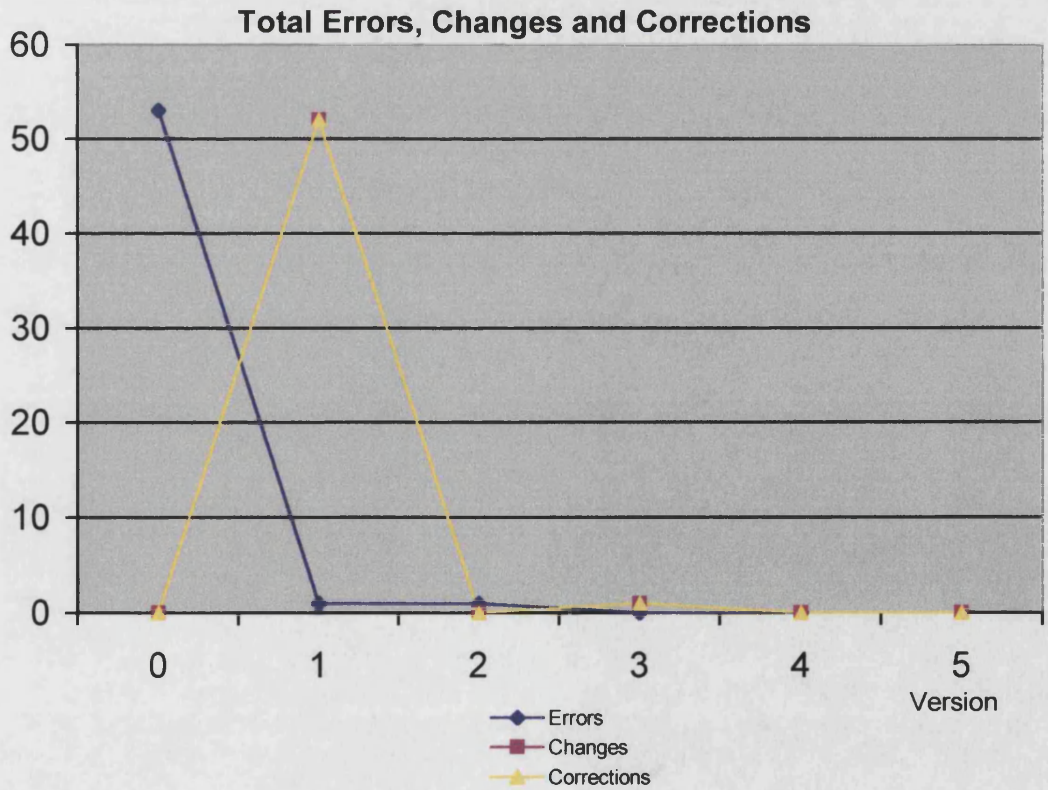


Figure 5.8 Results for model 4 (b)

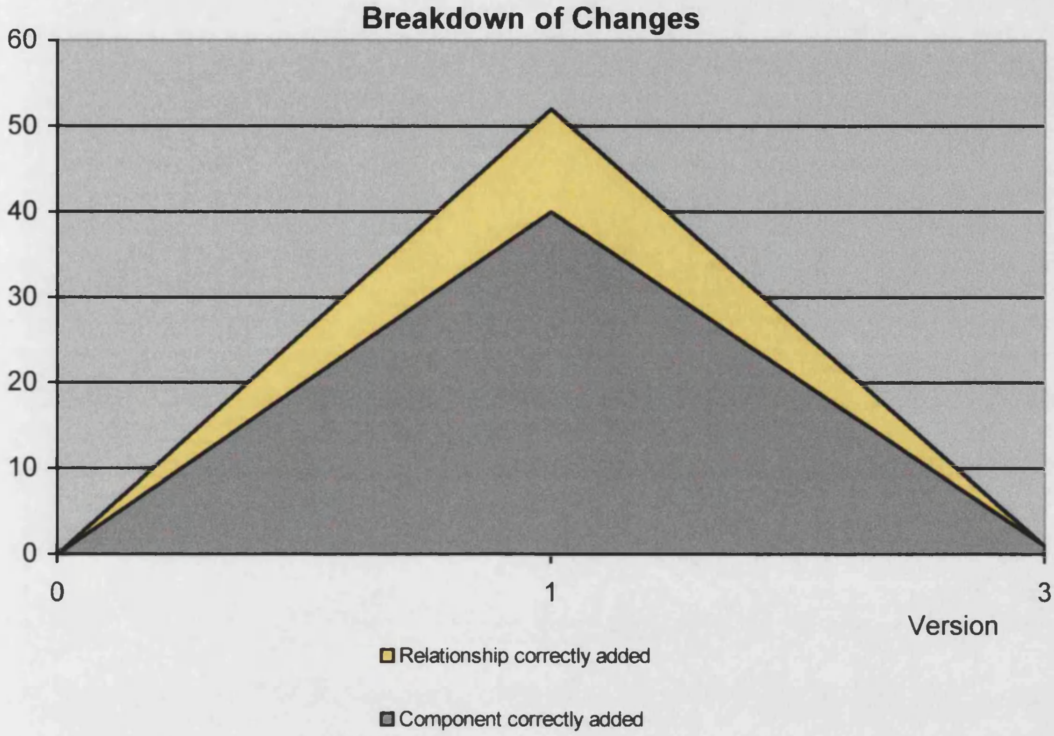


Figure 5.9 Results for model 4 (c)

5. Fund Management Model

The modeller in this study was a fourth-year computer science degree student with no experience of commercial systems development and no prior knowledge of the relevant business area (investment banking or funds management). Her main source of information was a contact in a US bank who was based in Switzerland and available only infrequently. In the absence of first-hand information the modeller resorted to using background information of a general nature that was available in books and sales literature, but had difficulty in relating this information to the particular situation being modelled. The modeller initially made many errors, adding far more incorrect components and relationships than correct ones. The first version of the model was therefore rather incorrect, but after this point completeness and correctness improved smoothly. Most of the work after version 1 consisted of fixing the initial mistakes. The modeller made some specific modelling errors repeatedly:

- Enumerating categories. For example, for the category 'Fund type' the modeller created a number of attribute components corresponding to different types of fund.

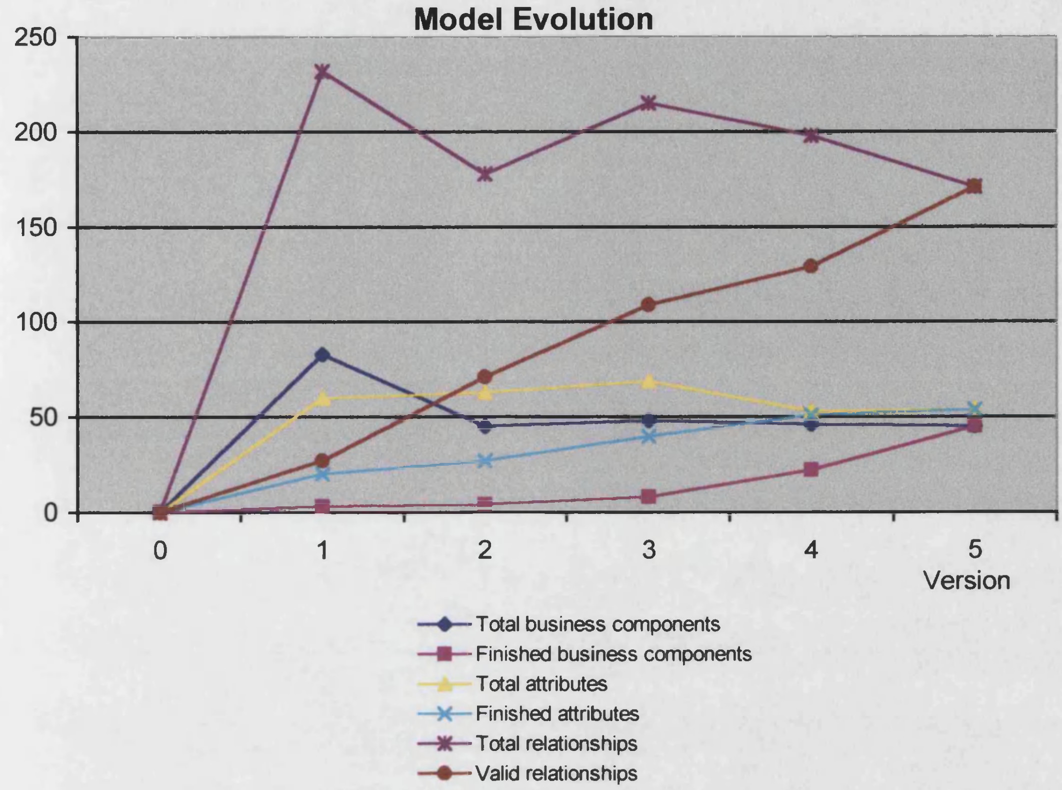
- Compensating for lack of knowledge by modelling in a non-specific way. The intention was to make the model more specific when further information became available. However, this also had the effect of over-complicating the model, making it harder to correct.
- Including multiple redundant relationships to "one-only" concepts (e.g. the subject company), resulting in further over-complexity.

A final version of the model was produced, correcting these errors (version 5). It is clear that the modeller experienced a dual 'learning curve'—learning about the business in question and also learning about the modelling technique. Her initial enthusiasm led her to model in the absence of knowledge, and hence the model she produced at first was significantly incorrect. But by the time the model had been completed, she had largely got to grips with both the subject matter and the modelling technique. The modeller made good use of embedded annotations, including source material from the subject organisation (PowerPoint presentations, Word documents, etc.). The model was large and a large number of changes were made overall. It was not possible to assess productivity for this model since modelling time was not recorded.

No of components	99	Changes per component	10.63
No of relationships	171	Errors per change	0.42
Relationships per component	3.80	No. of changes	1379
Attributes per component	1.20		

Breakdown of changes	Total	%
Relationship correctly removed	371	26.90
Incorrect relationship added	369	26.75
Relationship correctly added	177	12.83
Incorrect component added	129	9.35
Component correctly removed	129	9.35
Component correctly added	125	9.06
Component type corrected	21	1.52
Cardinality corrected	19	1.37
Component wrongly removed	13	0.94
Relationship wrongly removed	11	0.79
Component type changed wrongly	10	0.72
Cardinality altered wrongly	5	0.36

Table 5.5 Results for model 5



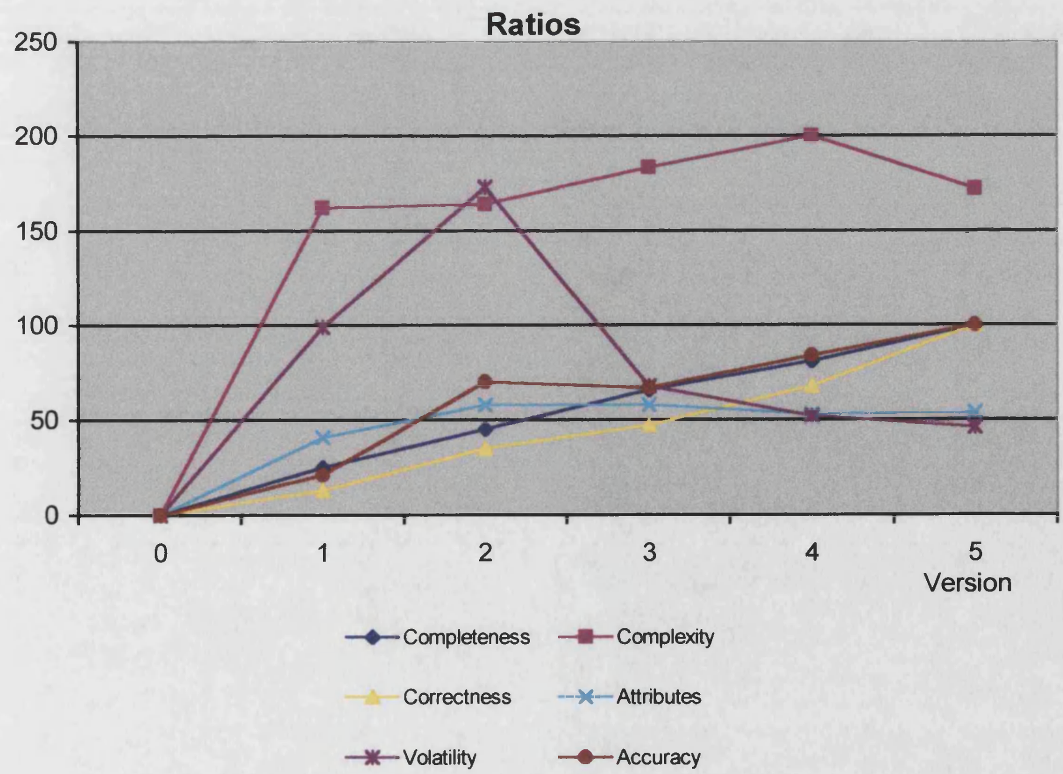
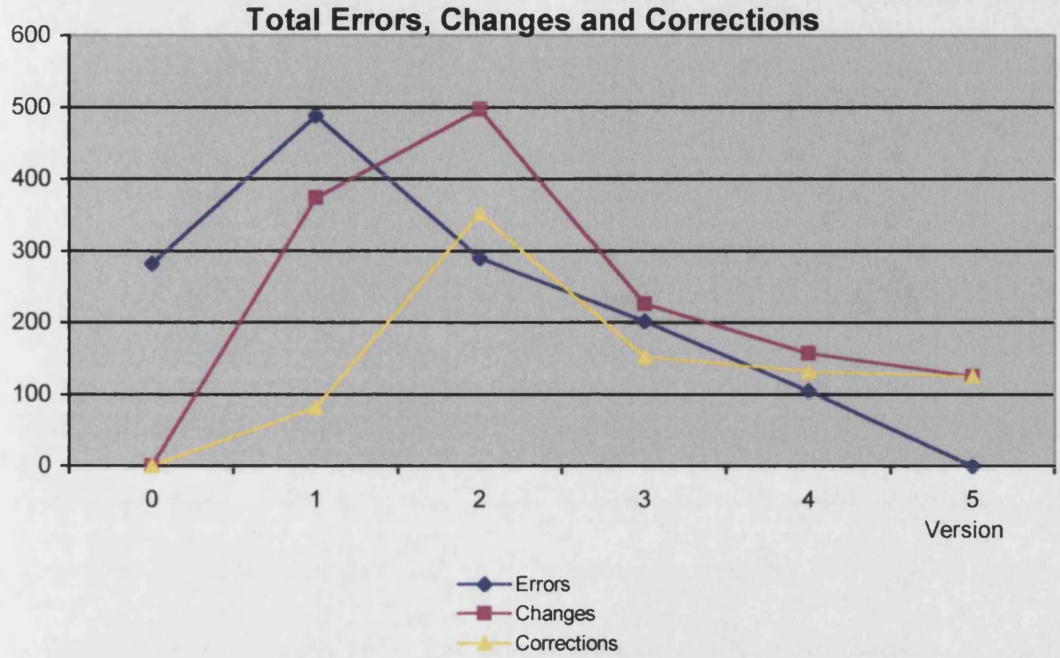


Figure 5.10 Results for model 5 (a)



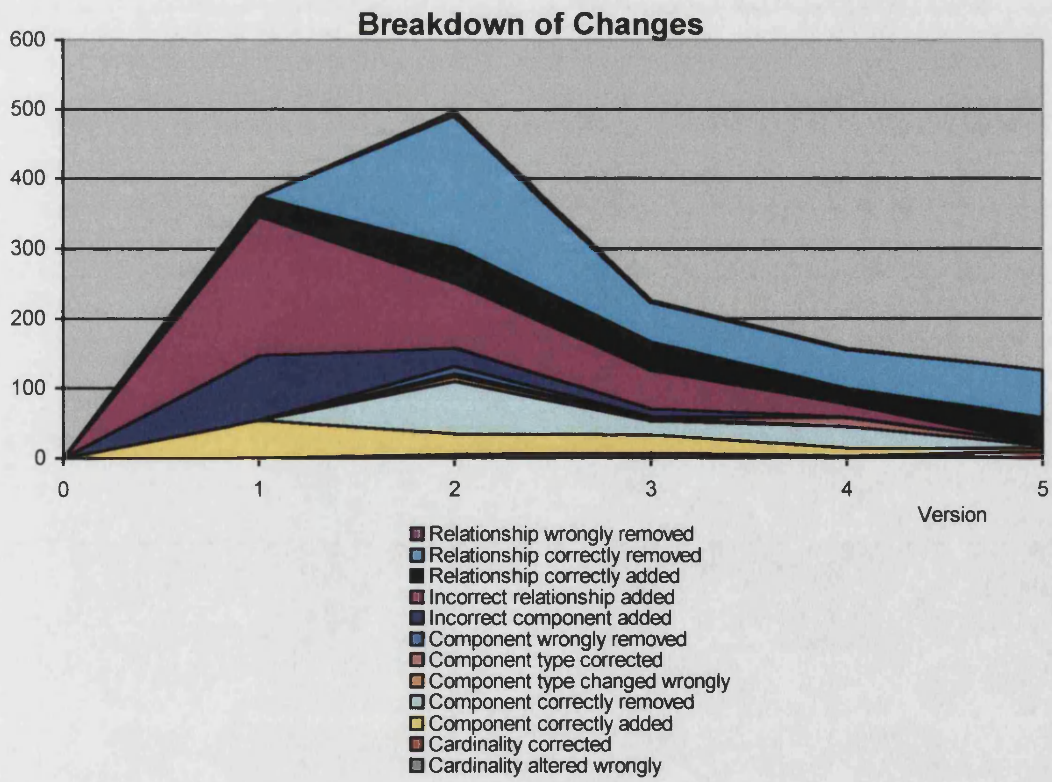


Figure 5.11 Results for model 5 (b)

6. Help Desk Model

This model was constructed by a project manager with experience of help desk systems. The modeller had no prior experience of object or data modelling but had previously used process modelling techniques and could claim some expertise in this area. This model is remarkable in that it exhibits a very strong 'mid-life crisis' in which complexity and error count both increase dramatically (versions 6 to 9). The modeller explained that she was exploring an idea for restructuring the model, based on her prior experience with other modelling techniques. The idea did not work, and so the modeller reverted to the original structure before completing the model. As a consequence of this the error rate for the model was rather inflated. However, productivity was reasonably good, and the upward trends in completeness and correctness were relatively unaffected. Apart from the failed experiment, modelling accuracy improved throughout. A corrected version was produced (version 17).

No of components	28	Changes per component	5.57
No of relationships	40	Errors per change	0.34
Relationships per component	4.44	Components per hour	2.40
Attributes per component	2.11	No. of changes	194

Breakdown of changes	Total	%
Relationship correctly removed	45	23.19
Incorrect relationship added	44	22.68
Relationship correctly added	37	19.07
Component correctly added	30	15.46
Incorrect component added	9	4.63
Component correctly removed	9	4.63
Cardinality corrected	8	4.12
Relationship wrongly removed	4	2.06
Cardinality altered wrongly	4	2.06
Component wrongly removed	2	1.03
Component type corrected	1	0.51
Component type changed wrongly	1	0.51

Table 5.6 Results for model 6

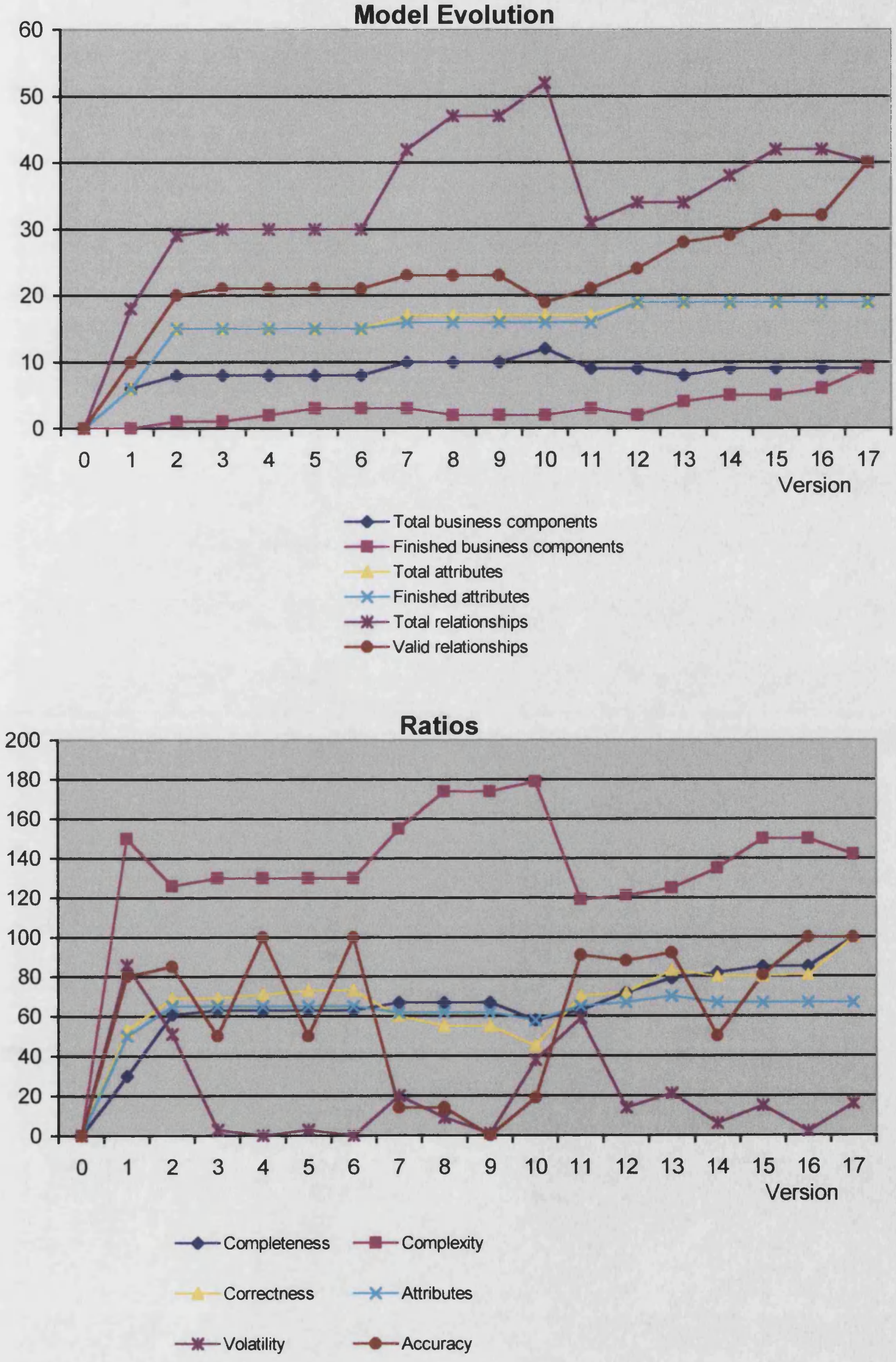
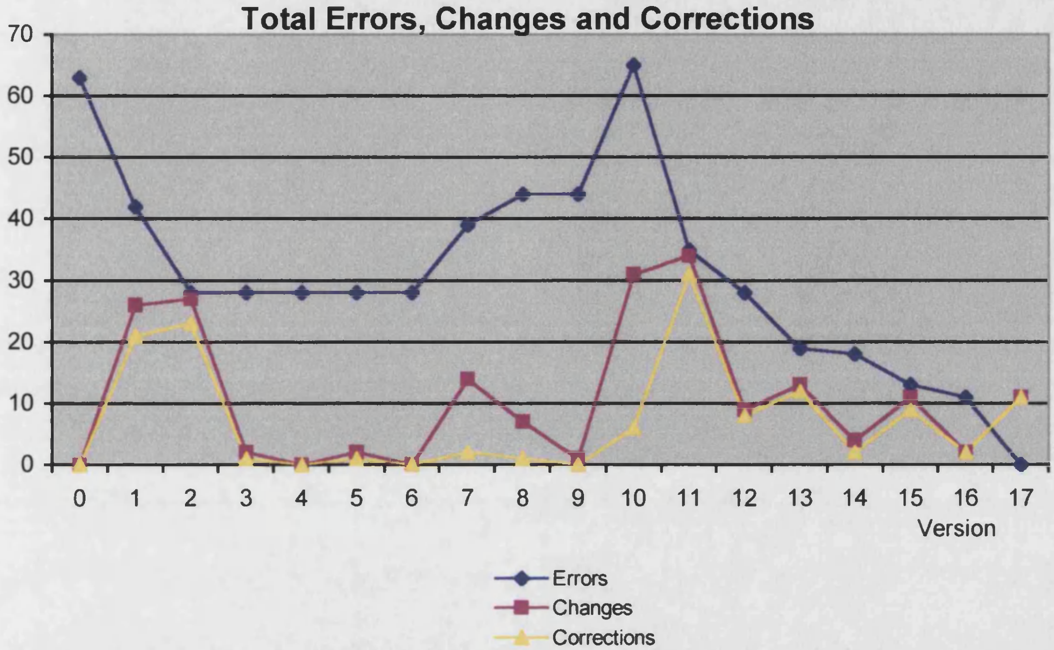


Figure 5.12 Results for model 6 (a)



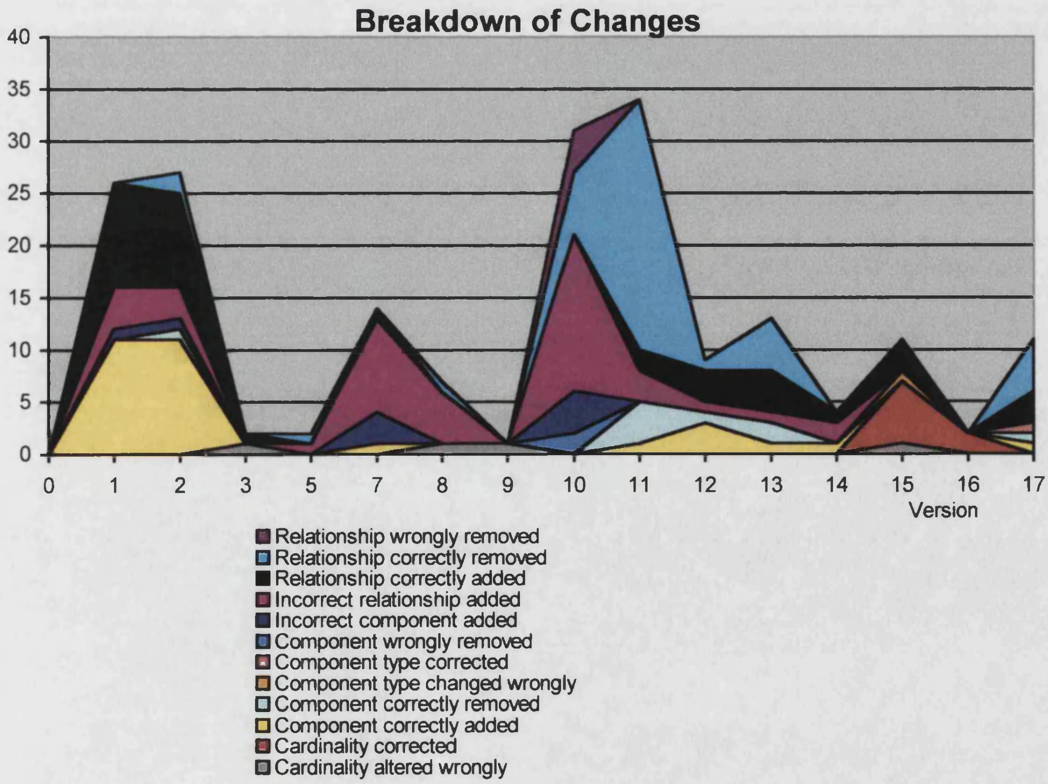


Figure 5.13 Results for model 6 (b)

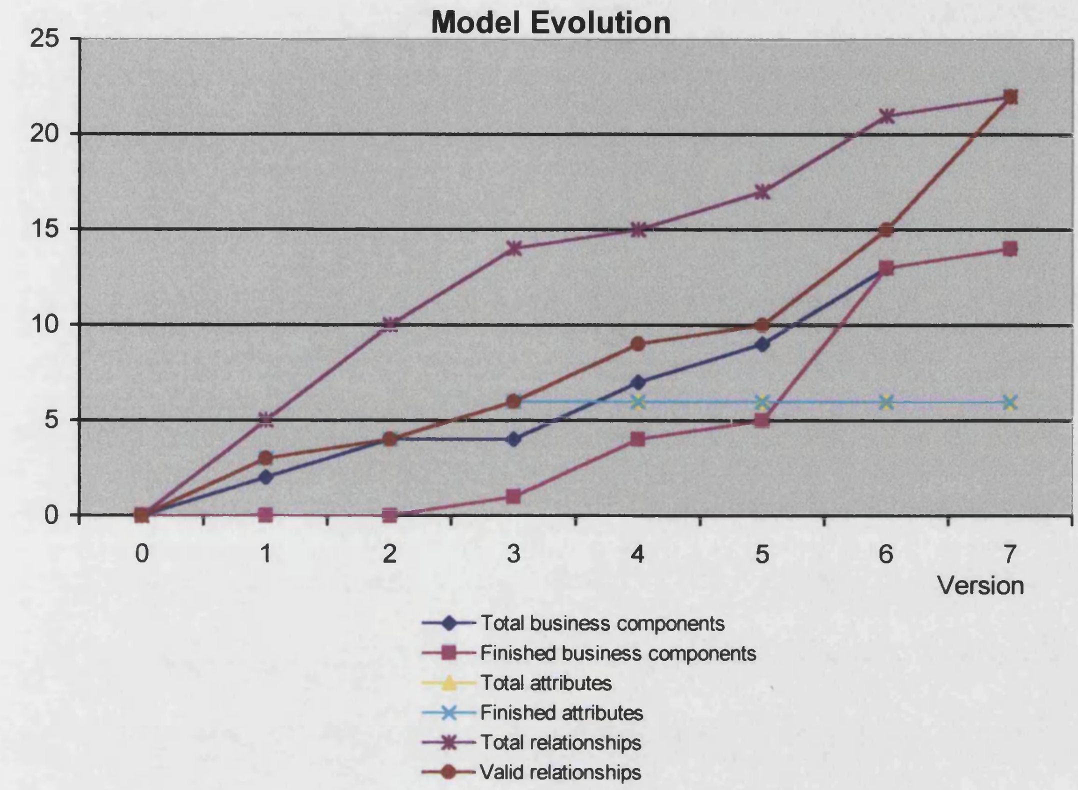
7. Homeopathic Remedies Model

This small model was produced by the administrator in a homeopathic medical practice. The modeller had limited experience of business, made modest use of computers (mainly for word processing) and had no experience of software development or modelling. The model is unusual in that it consists largely of abstract categories, which are related in a number of different ways. The model is about homeopathic knowledge (the "Materia Medica"—symptoms and remedies) rather specific treatments administered to specific patients. The homeopathic Materia Medica is a large and complex body of information and, despite a superficial appearance of systematic organisation, it is not easy to see how its underlying structure might be codified. The main difficulty stems from the informality and overlapping nature of the abstract classifications used for remedies and symptoms: aetiologies, modalities, physical generals, physical particulars, "strange, rare or peculiar", and characteristics. The abstract nature of the subject matter was hard for the modeller to conceptualise, despite a good knowledge of homeopathic practice. Hence this was considered a difficult area to model. As a result the modeller probably spent more time than most modellers in puzzling out how to structure the model, and its relatively small size reflects this fact rather than any lack of effort or thought on the part of the modeller. Nevertheless, the total number of errors declined smoothly and the error rate was low overall. The modeller appeared to grasp the principles of modelling quite quickly. Completeness and correctness improved steadily and productivity was high. A corrected version was produced (7).

No of components	20	Changes per component	2.75
No of relationships	22	Errors per change	0.14
Relationships per component	1.57	Components per hour	4.33
Attributes per component	0.42	No. of changes	55

Breakdown of changes	Total	%
Component correctly added	20	36.36
Relationship correctly added	19	34.54
Relationship correctly removed	7	12.72
Incorrect relationship added	5	9.09
Incorrect component added	2	3.63
Component correctly removed	2	3.63

Table 5.7 Results for model 7



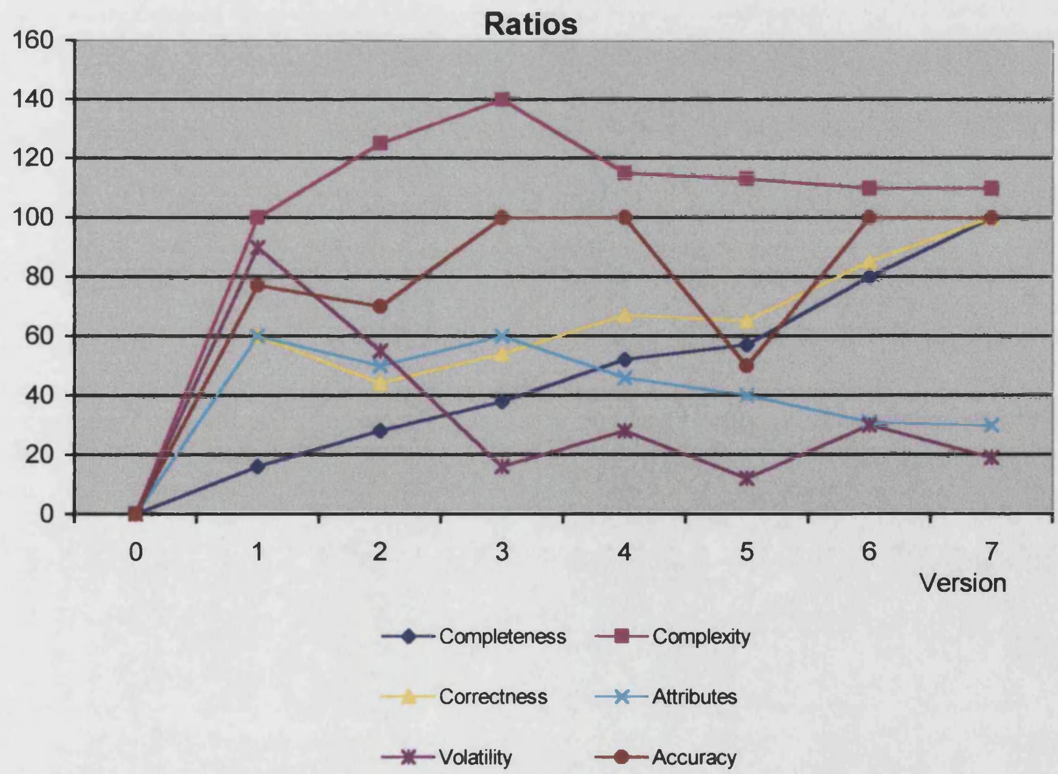
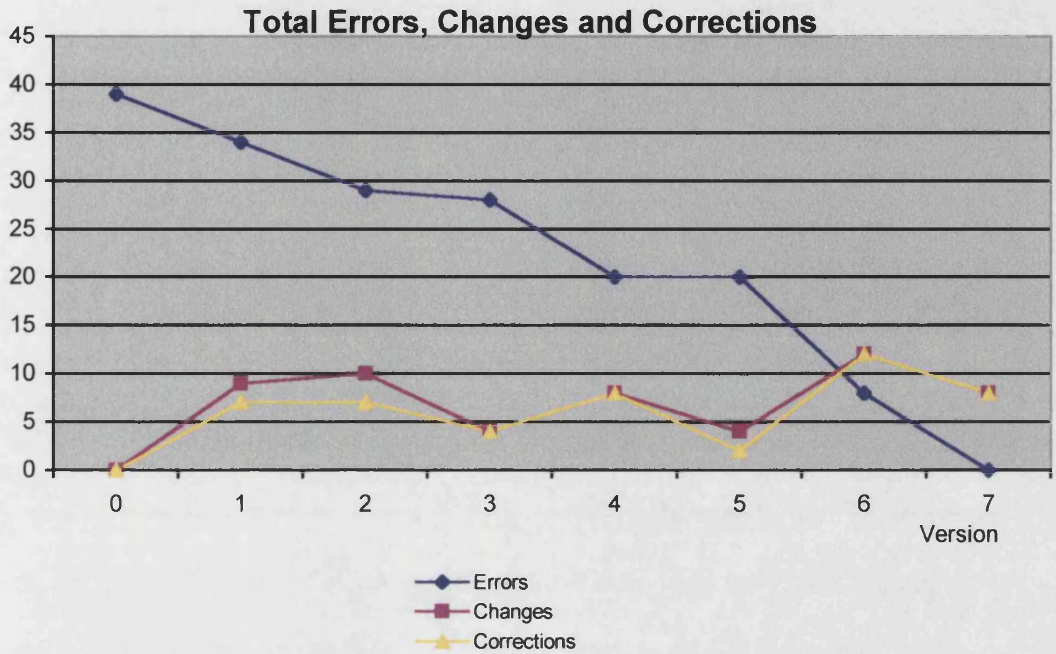


Figure 5.14 Results for model 7 (a)



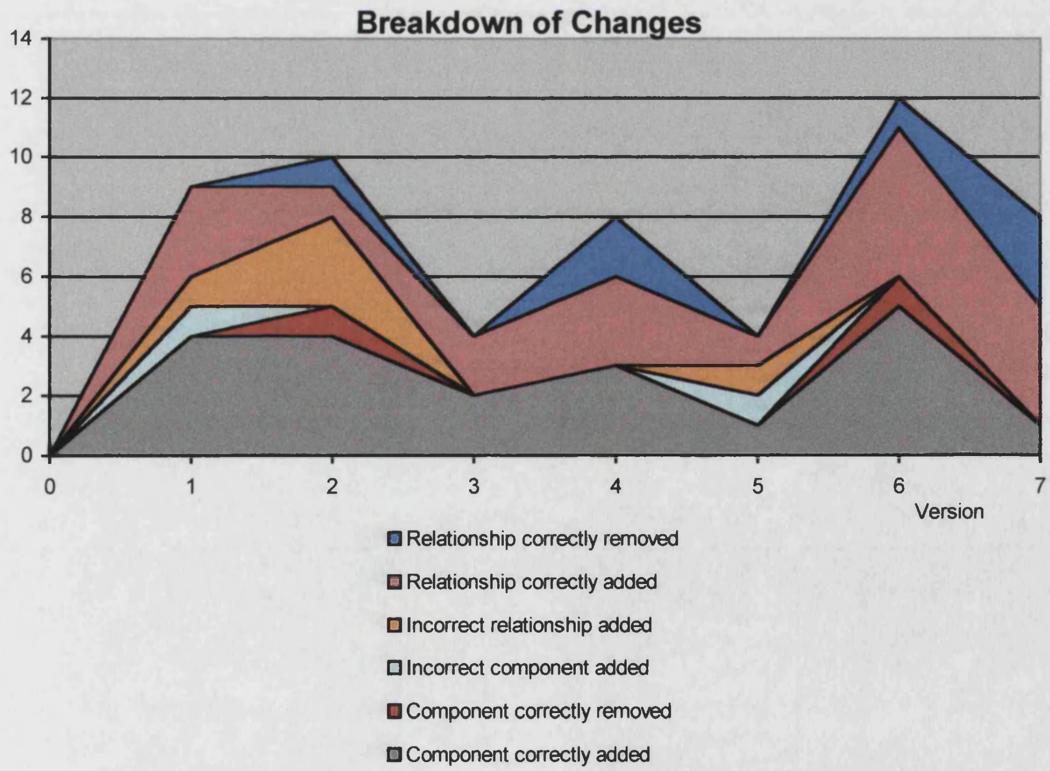


Figure 5.15 Results for model 7 (b)

8. Human Resources Model

This model was produced by a company administrator who is expert in human resources but inexperienced with computers and wholly ignorant of software development techniques. The modeller was enthusiastic about the modelling task and set about producing a quick succession model of versions, each different from its predecessor. Despite the level of activity, overall productivity was poor and the error rate was high. The earlier models appeared to resemble conventional object model diagrams, but the modeller soon began to introduce notations of her own. It is not clear that she understood the object modelling notation and it is possible that she did not fully appreciate the general idea that models could be interpreted formally. It was clear that the idea of a diagram as a set of logical propositions was an unfamiliar one. The modelling style was 'impressionistic' throughout (meaning that the modeller could interpret the model, but not in a formal or definitive way and not according to any standard notation). The modeller paid little attention to correcting the meaning of the model, and relationships were added indiscriminately. Although the number of errors in the model increased steadily, the modeller remained unaware of them. She expressed the belief that the model represented the business of human resources well, and felt that her notation conformed to method 'Y' as outlined during the introduction. At no stage were errors in the model recognised. The final (uncorrected) version of the model was over-complex and contained many incorrect relationships (there were 30% too many relationships overall, and 90% of all relationships were incorrect). A corrected version was produced (version 7), with significantly reduced complexity, but using essentially the same components.

No of components	79	Changes per component	7.59
No of relationships	96	Errors per change	0.63
Relationships per component	4.36	Components per hour	1.00
Attributes per component	2.59	No. of changes	619

Table 5.8 Results for model 8 (a)

Breakdown of changes	Total	%
Relationship correctly removed	160	25.84
Incorrect relationship added	160	25.84
Relationship correctly added	104	16.80
Component correctly added	81	13.08
Incorrect component added	49	7.91
Component correctly removed	49	7.91
Relationship wrongly removed	10	1.61
Component wrongly removed	2	0.32
Cardinality corrected	2	0.32
Component type corrected	1	0.16
Component type changed wrongly	1	0.16

Table 5.8 Results for model 8 (b)

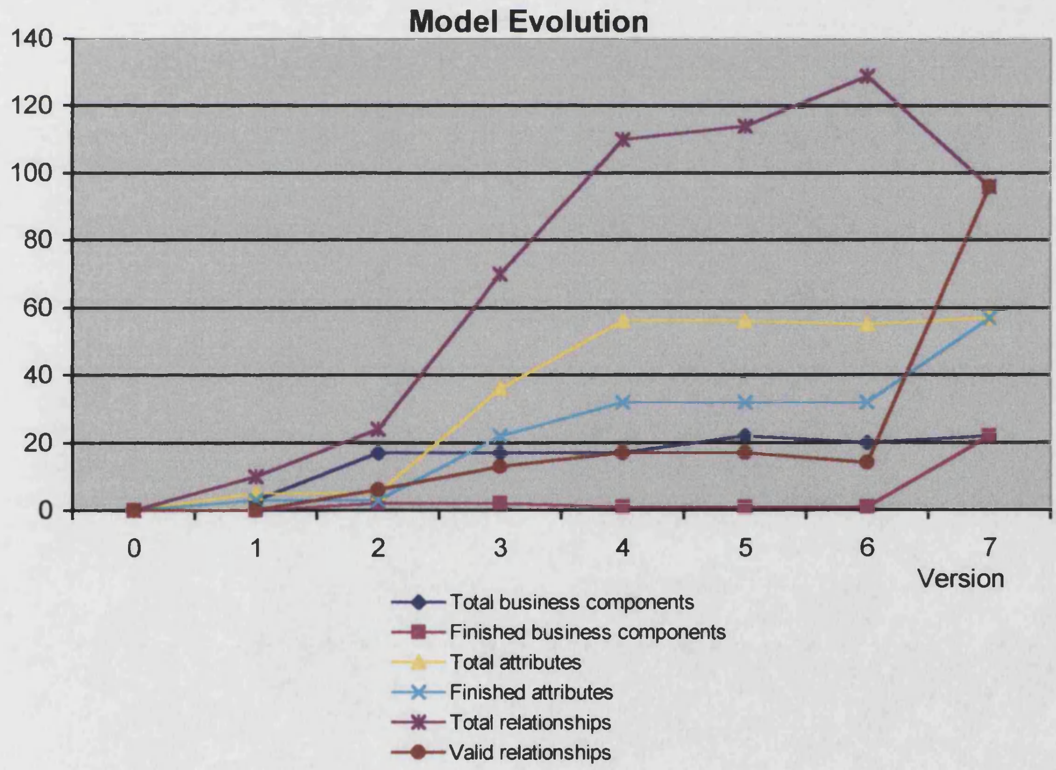
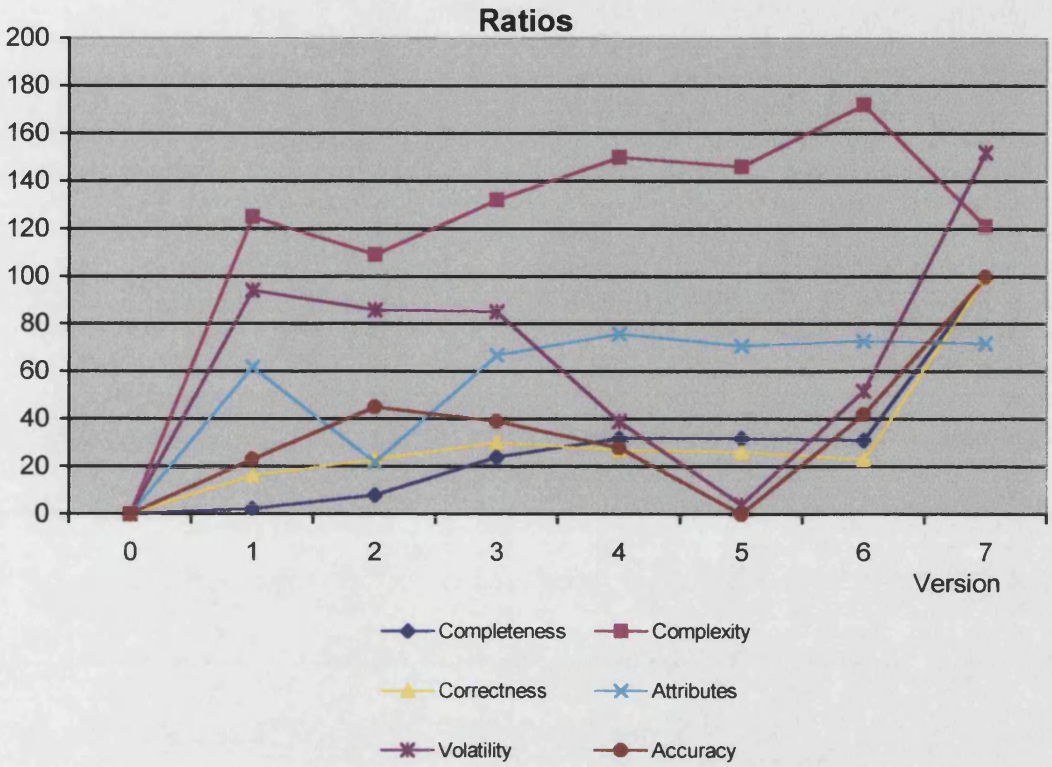


Figure 5.16 Results for model 8 (a)



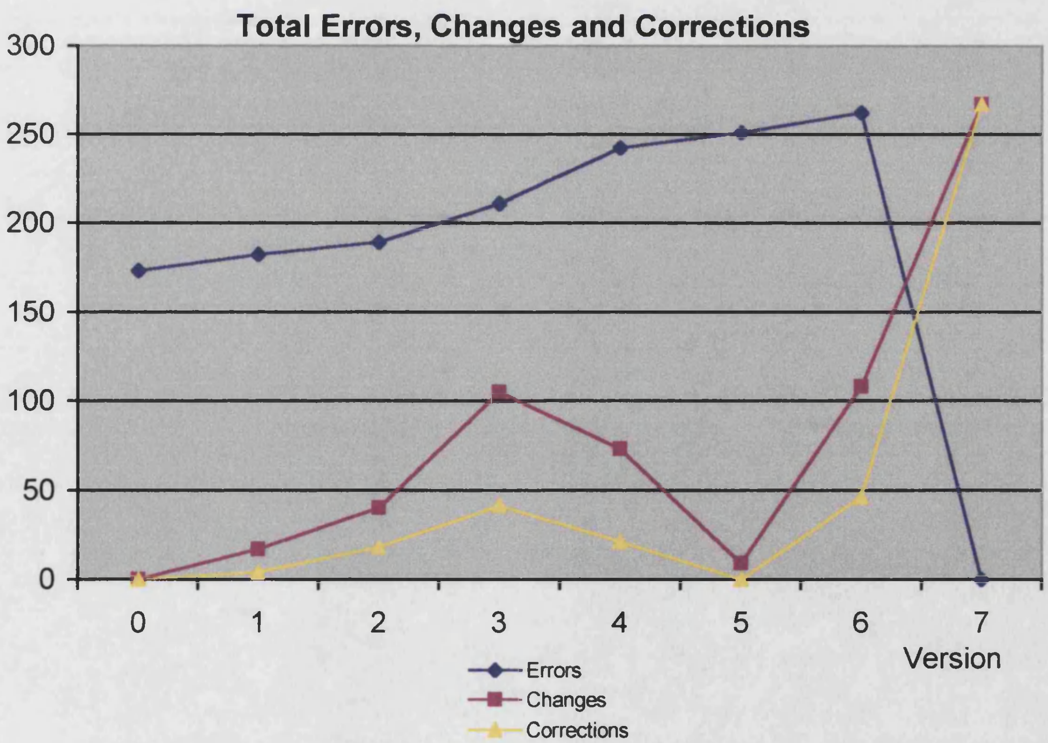


Figure 5.17 Results for model 8 (b)

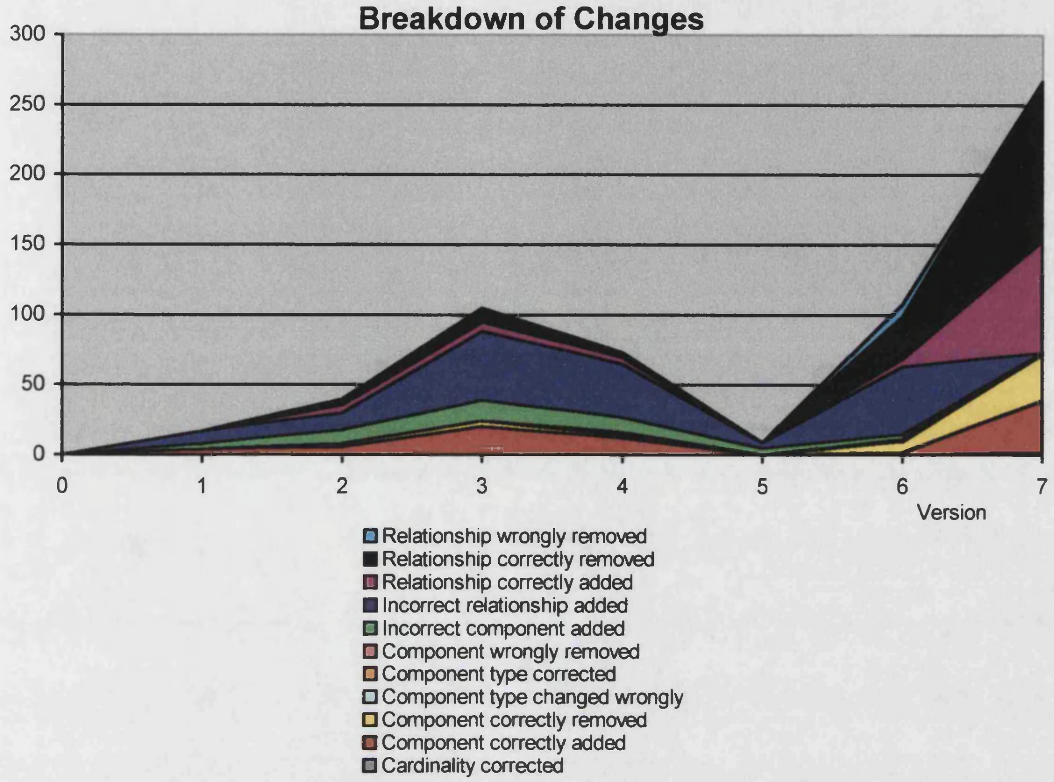


Figure 5.18 Results for model 8 (c)

9. International Roaming Model

This larger model was produced by a trained data modeller who has gained many years of experience in systems analysis (and modelling techniques in particular). Before starting work, the modeller expressed doubts about the workability of method 'X'. One objection was the fact that the technique did not allow use of a flipchart and, therefore, could not produce large wall-charts. The modeller was in the habit of taping models to the wall during modelling sessions, in order to remind participants of their progress and to help motivate them. A second objection concerned the fact that method 'X' allowed associations between concepts to remain unspecified. Being accustomed to traditional modelling notations, the modeller felt this was an unsafe situation and preferred to define every association explicitly.

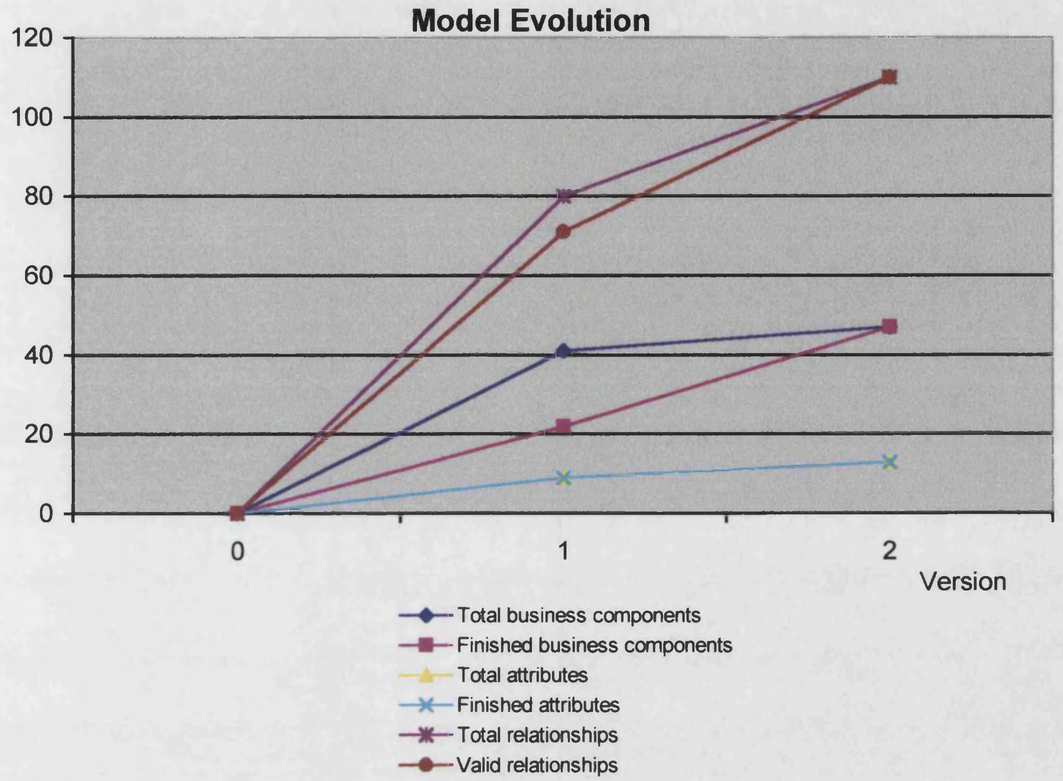
After using method 'X' for two or three sessions, the modeller became very enthusiastic about the modelling technique, to the extent of wanting to abandon use of method 'Y' altogether in the other parts of this study. He was persuaded to continue use of method 'Y', however. This modeller made extensive use of coloured backgrounds and eye-catching graphics to represent model components (for example, working group objectives were represented using a picture of Moses holding the ten commandments). He also populated the model with a good number of annotations that were directly relevant to the modelling sessions. The annotations were used during the modelling session and helped to remind the group of what topics they were to address. Despite not having used the modelling technique before, the modeller made relatively few errors and productivity was high. No corrected version was necessary.

The modeller had been trained in facilitation and believed strongly in the importance of group motivation and cohesion. To help explain the modelling technique, and to communicate his own enthusiasm, he prepared a small model called 'Jazz' that represented his own record collection. The model used a musical notation background and contained an embedded audio-visual clip of the singer Billie Holiday performing a well-known jazz song. The modeller used this model and, in particular, the embedded video clip, to illustrate the modelling tool in action and to give the group participants what he termed *"a visual jab in the ribs"*. This was an alternative to other ice-breaking and motivational exercises that he would normally have employed at the commencement of modelling sessions.

No of components	60	Changes per component	2.72
No of relationships	110	Errors per change	0.07
Relationships per component	2.34	Components per hour	3.91
Attributes per component	0.27	No. of changes	217

Breakdown of changes	Total	%
Relationship correctly added	102	47.00
Component correctly added	78	35.94
Relationship correctly removed	9	4.14
Incorrect relationship added	9	4.14
Incorrect component added	7	3.22
Component correctly removed	7	3.22
Cardinality corrected	4	1.84
Component type corrected	1	0.46

Table 5.9 Results for model 9



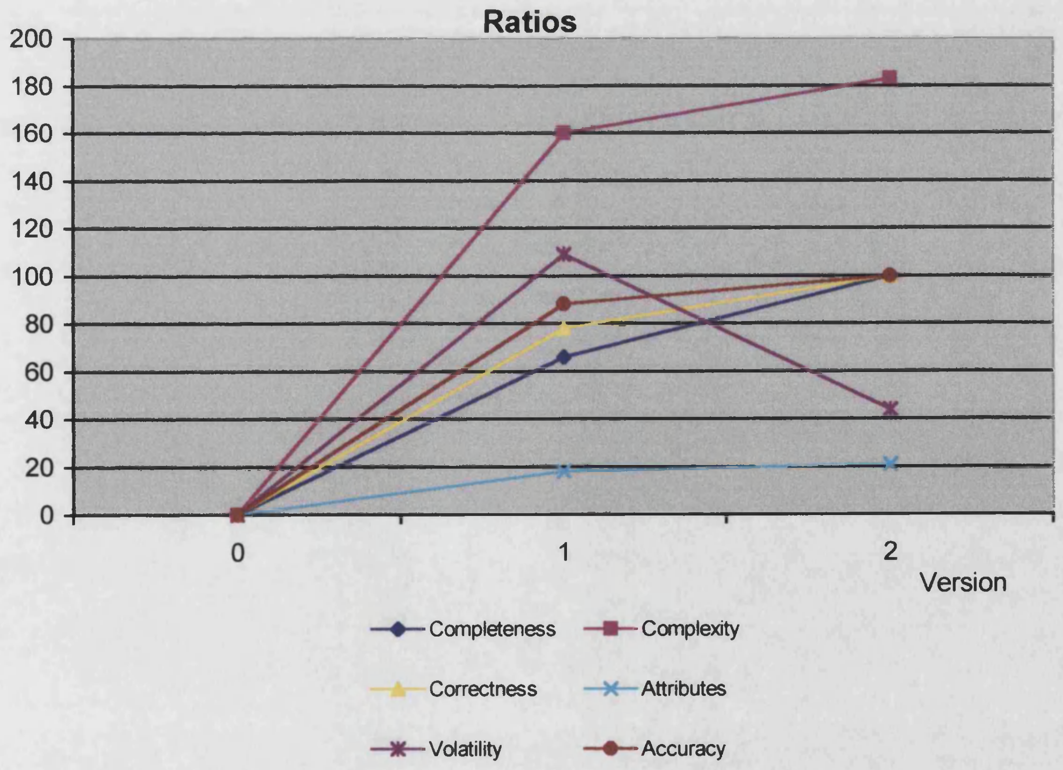
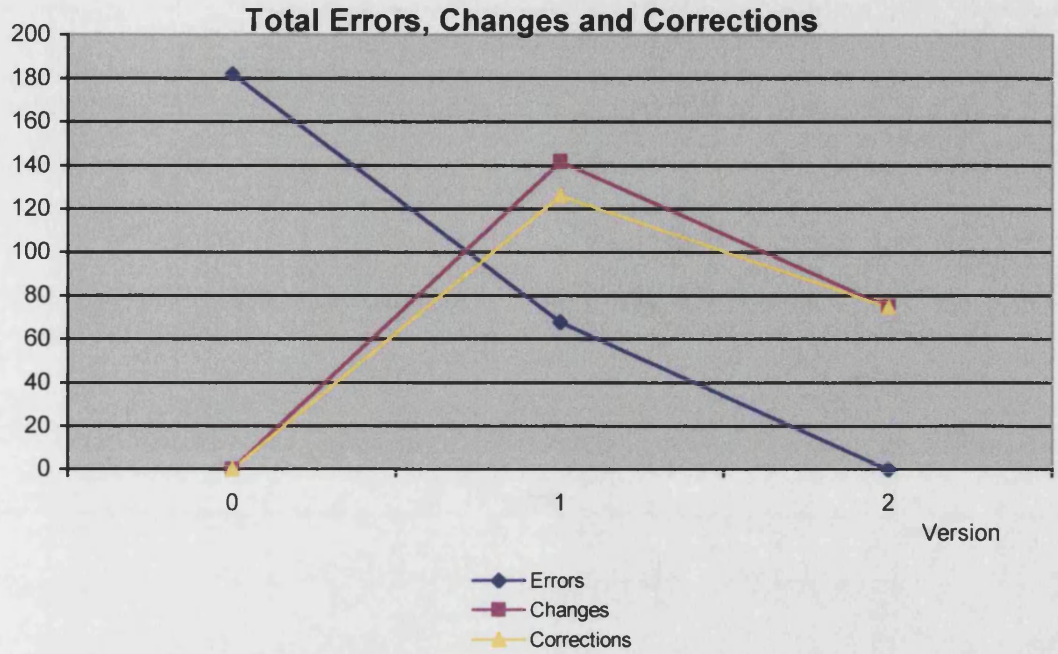


Figure 5.19 Results for model 9 (a)



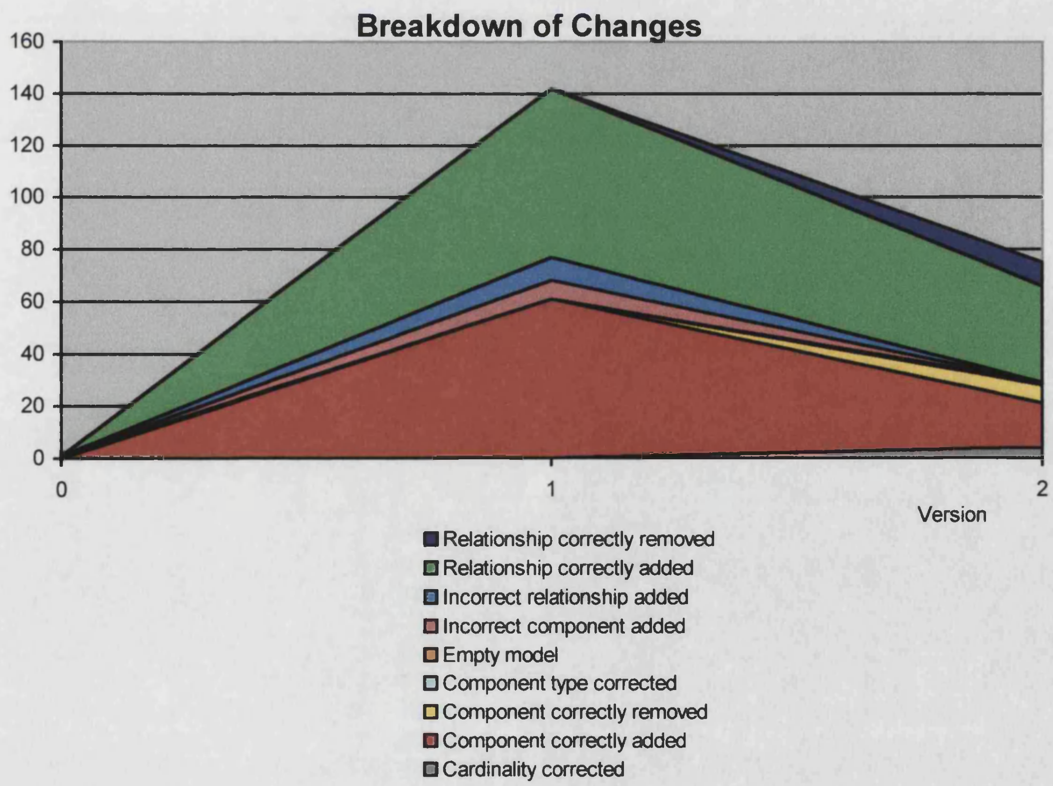


Figure 5.20 Results for model 9 (b)

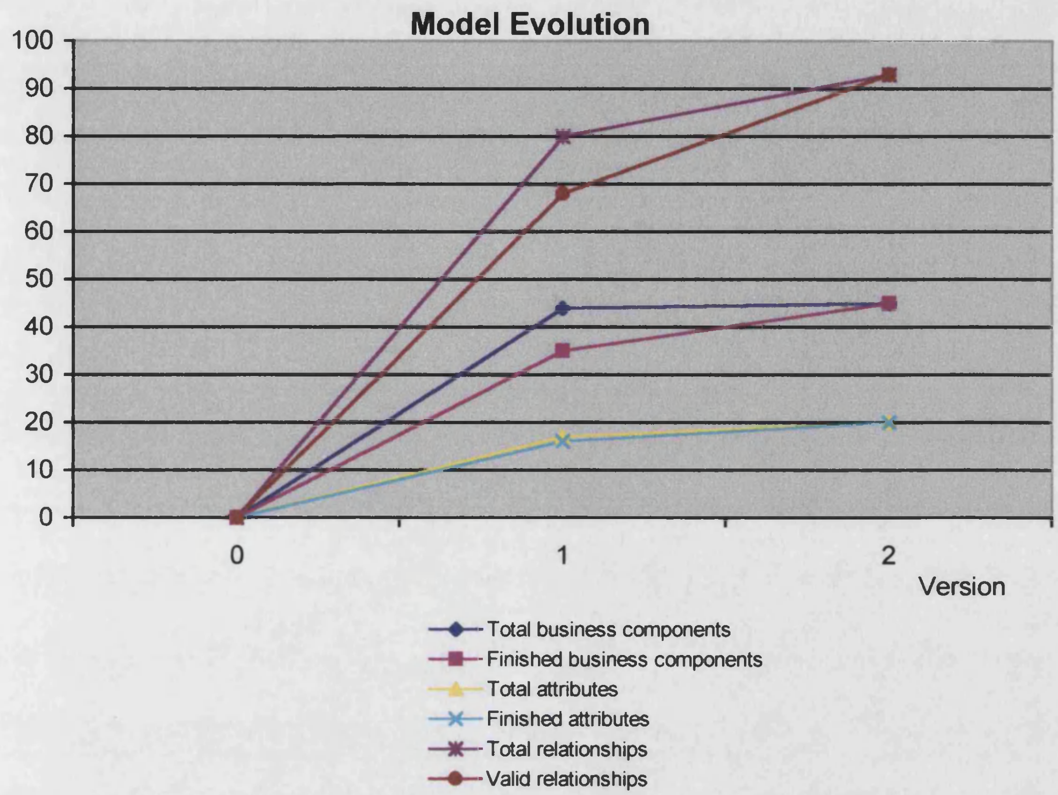
10. Legal Model

This model was produced by the same analyst as the previous model and follows a very similar pattern. Productivity was very high. In common with most of the models produced by this modeller, few mistakes were made and the model quickly became complete and correct. No corrected version was required.

No of components	65	Changes per component	2.52
No of relationships	93	Errors per change	0.08
Relationships per component	2.06	Components per hour	4.50
Attributes per component	0.44	No. of changes	201

Breakdown of changes	Total	%
Relationship correctly added	85	42.28
Component correctly added	78	38.80
Relationship correctly removed	12	5.97
Incorrect relationship added	12	5.97
Incorrect component added	5	2.48
Component correctly removed	5	2.48
Cardinality corrected	3	1.49
Component type corrected	1	0.49

Table 5.10 Results for model 10



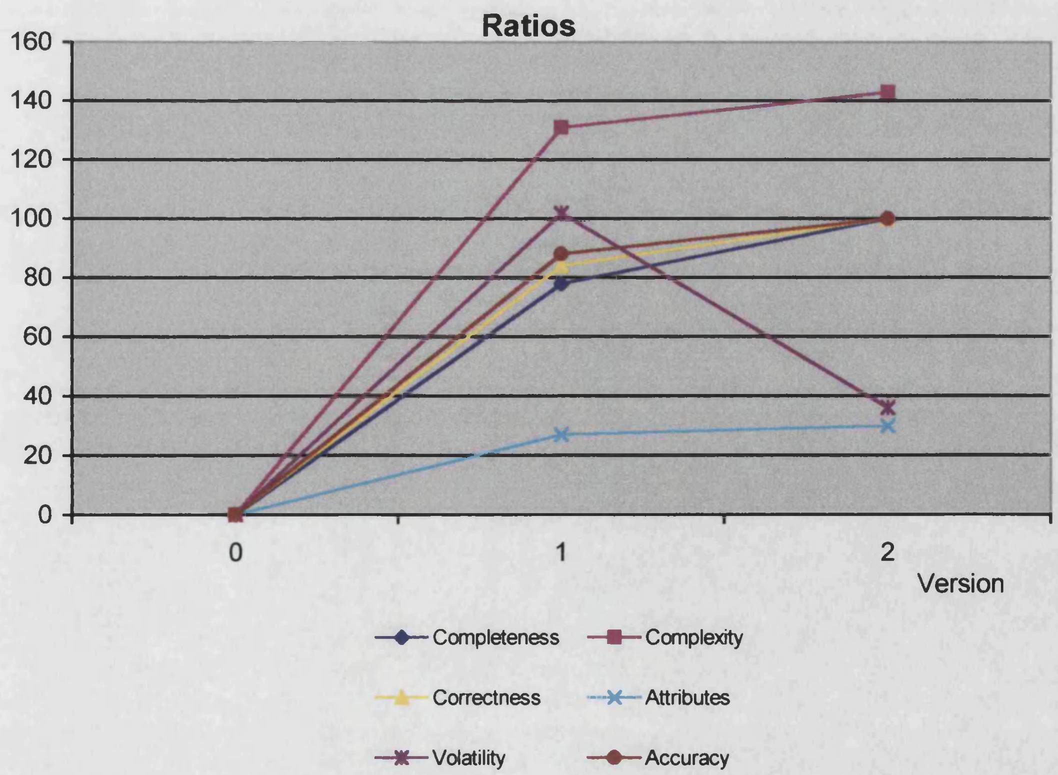
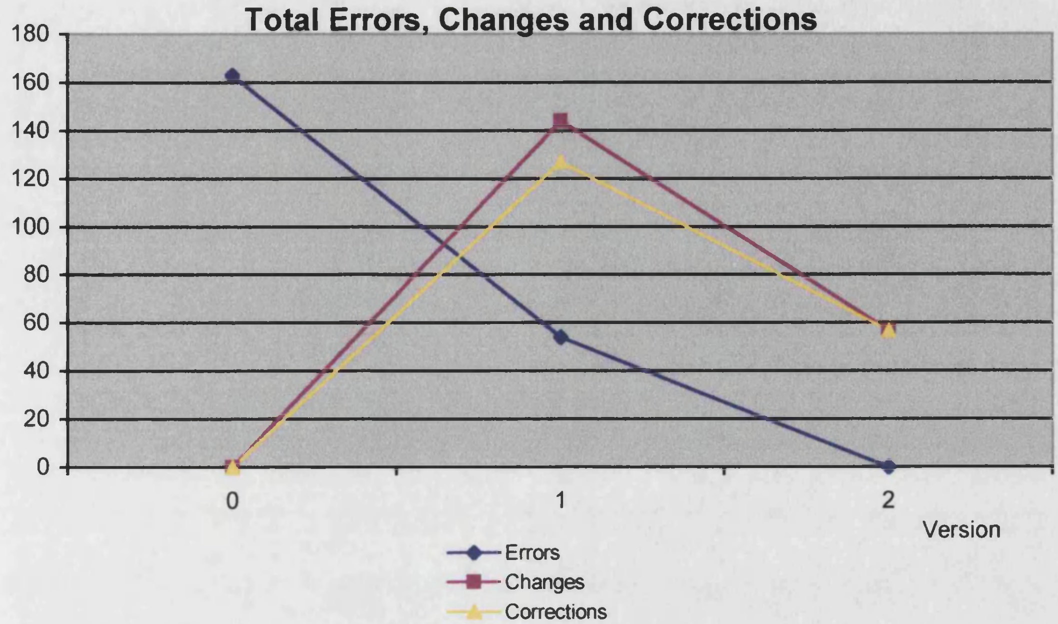


Figure 5.21 Results for model 10 (a)



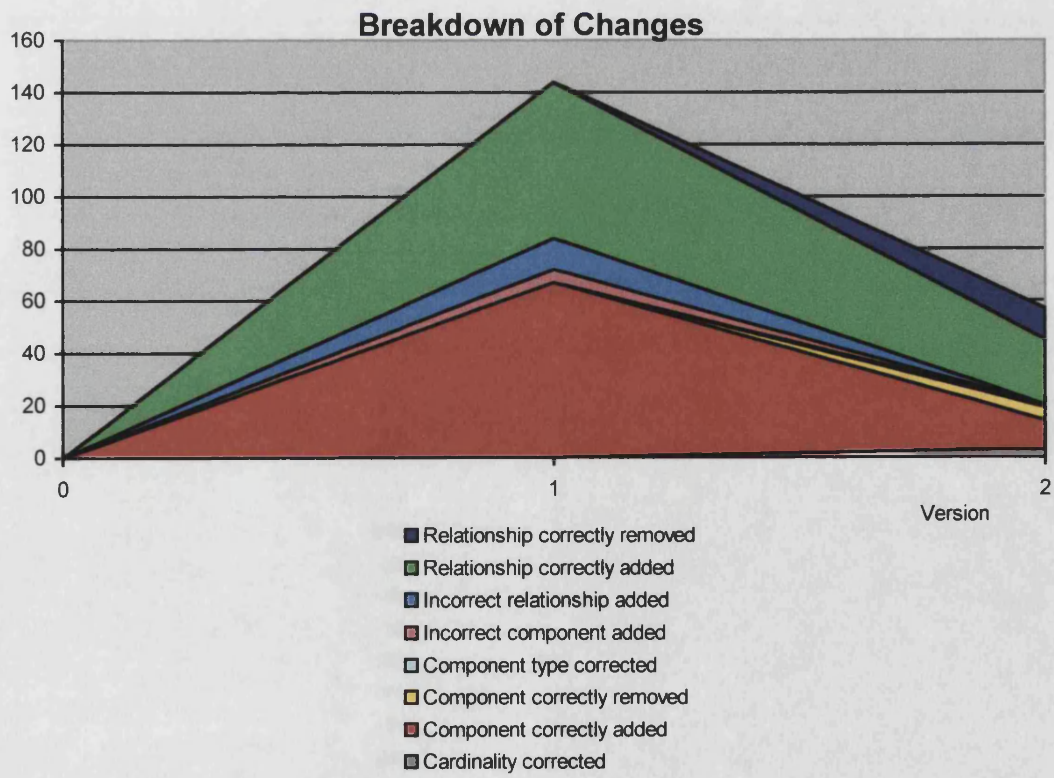


Figure 5.22 Results for model 10 (b)

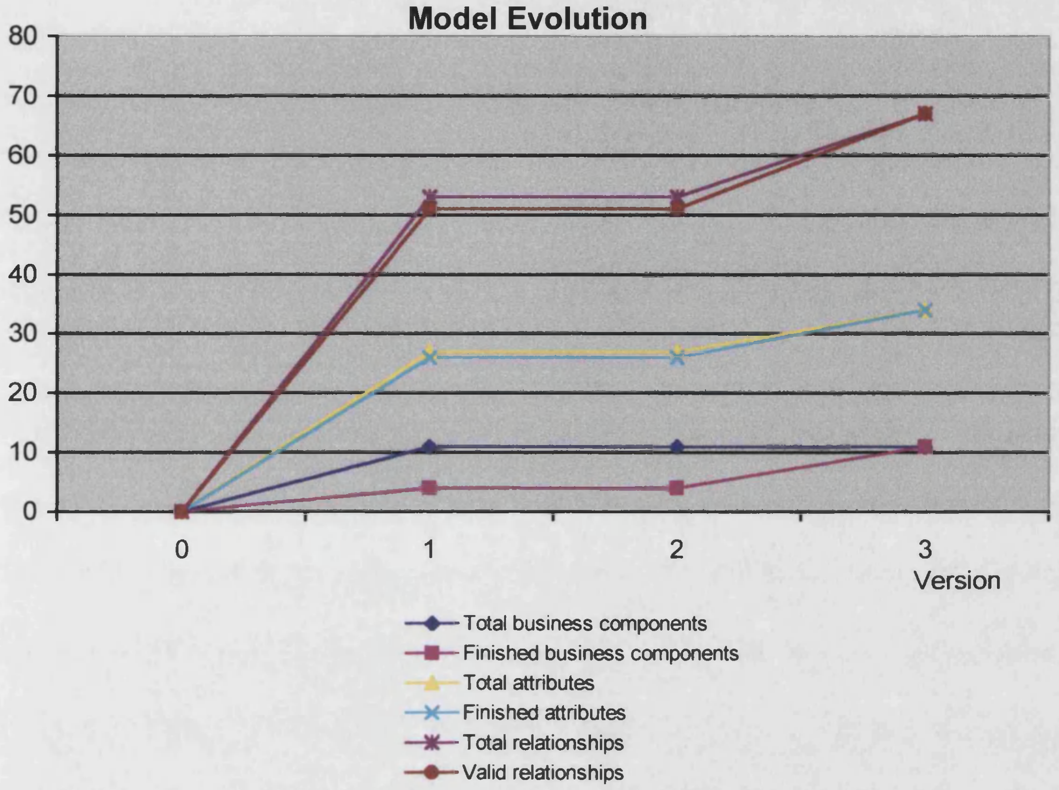
11. Legal and Regulatory Issues and Advice Model

This model was produced by the same experienced modeller as the previous two models. Once again, the modeller made few mistakes of any type. The model was 80% complete in the first version and revised only once after that. However, productivity was significantly worse than in the previous two models. The final quality of the model was judged to be sufficient that no corrected version was considered necessary.

No of components	45	Changes per component	2.37
No of relationships	67	Errors per change	0.02
Relationships per component	6.09	Components per hour	1.10
Attributes per component	3.09	No. of changes	107

Breakdown of changes	Total	%
Relationship correctly added	56	52.33
Component correctly added	45	42.05
Relationship correctly removed	2	1.86
Incorrect relationship added	2	1.86
Incorrect component added	1	0.93
Component correctly removed	1	0.93

Table 5.11 Results for model 11



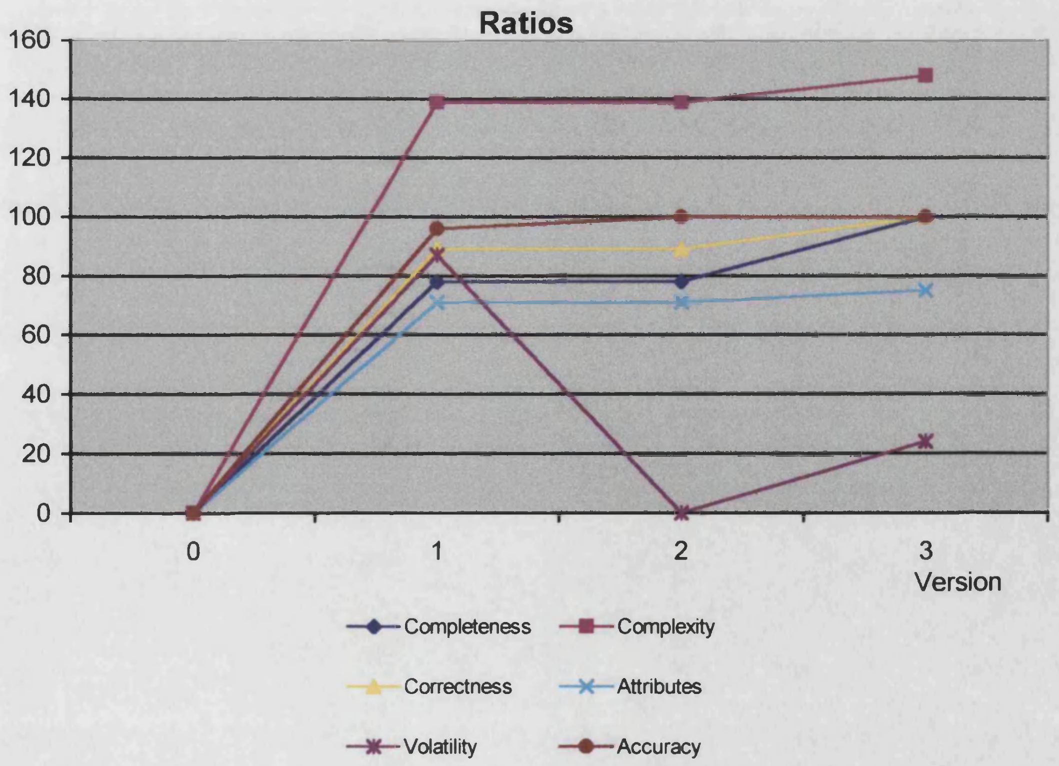
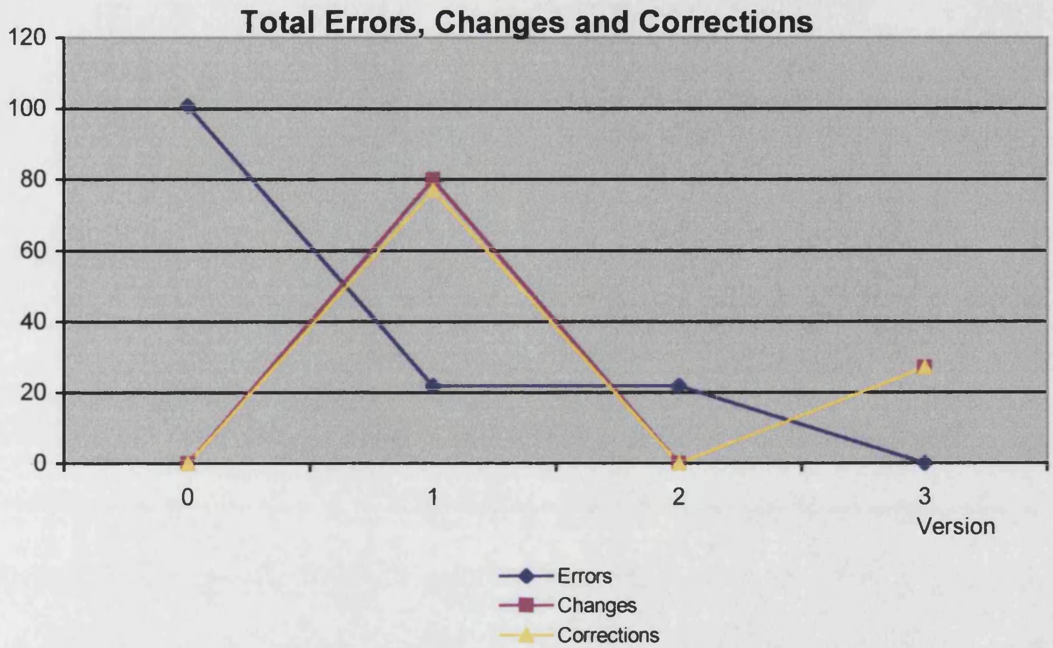


Figure 5.23 Results for model 11 (a)



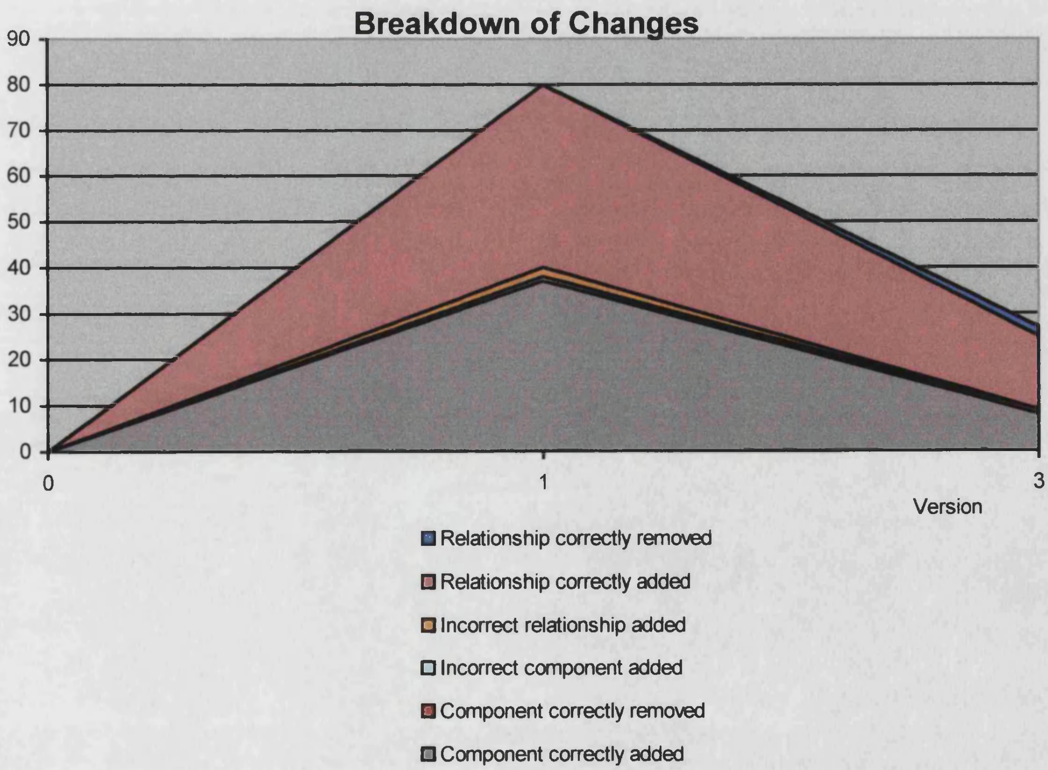


Figure 5.24 Results for model 11 (b)

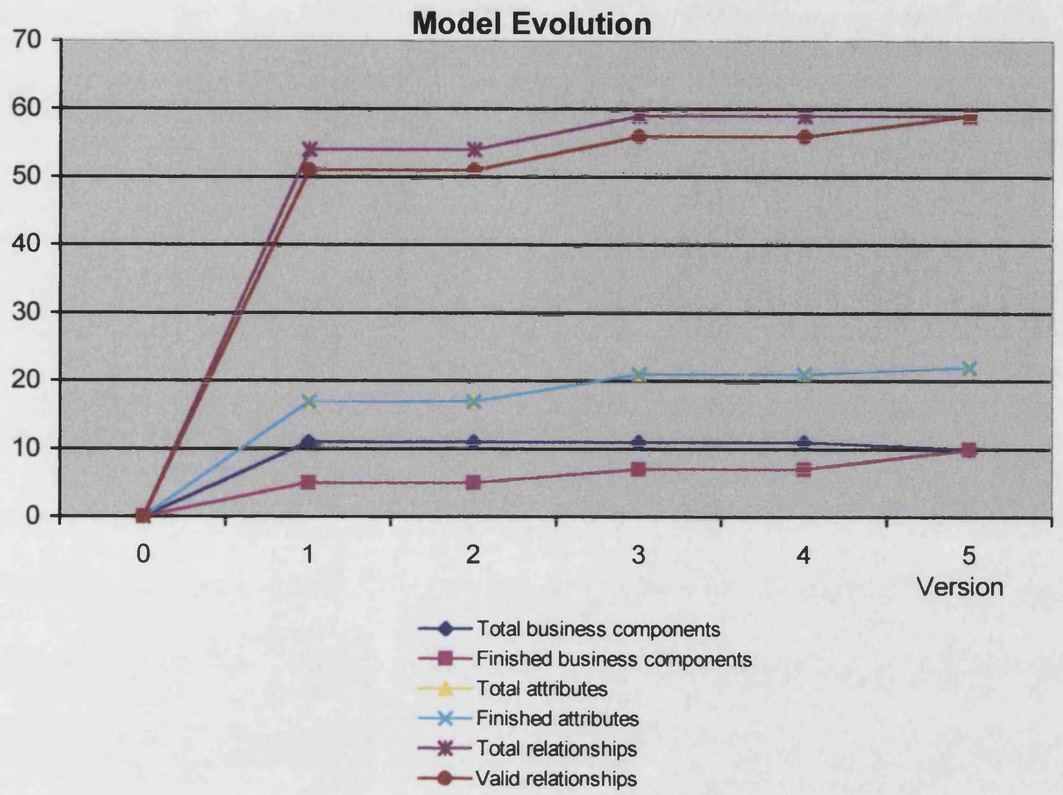
12. Mobile Phone Billing Model

This model was produced by the same modeller as the previous three models. Again, the modeller made very few errors. The model is relatively small and quickly converged on the finished version. The quality of the model was sufficient that no corrected version was considered necessary. Productivity (2.5 finished components per hour) was noticeably improved in this model over the previous model but still fell short of that obtained for models 9 and 10. The error rate was low.

No of components	32	Changes per component	2.49
No of relationships	59	Errors per change	0.03
Relationships per component	5.90	Components per hour	2.50
Attributes per component	2.20	No. of changes	87

Breakdown of changes	Total	%
Relationship correctly added	48	55.17
Component correctly added	32	36.78
Relationship correctly removed	3	3.44
Incorrect relationship added	3	3.44
Component type corrected	1	1.14

Table 5.12 Results for model 12



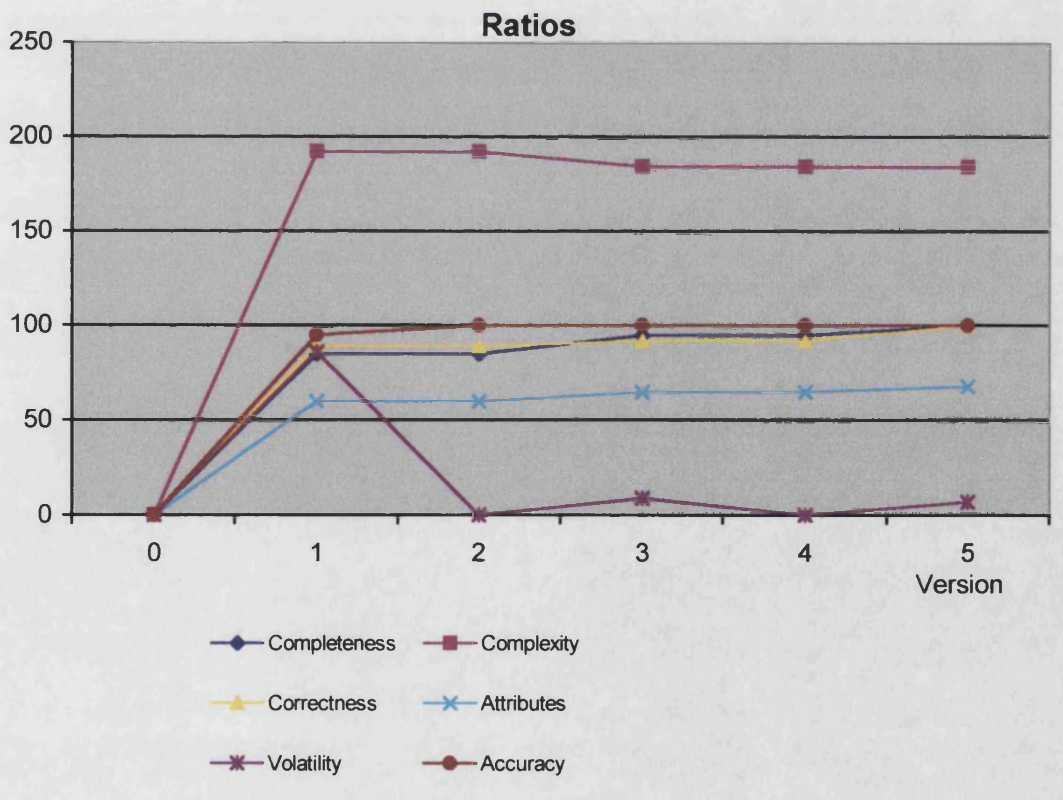
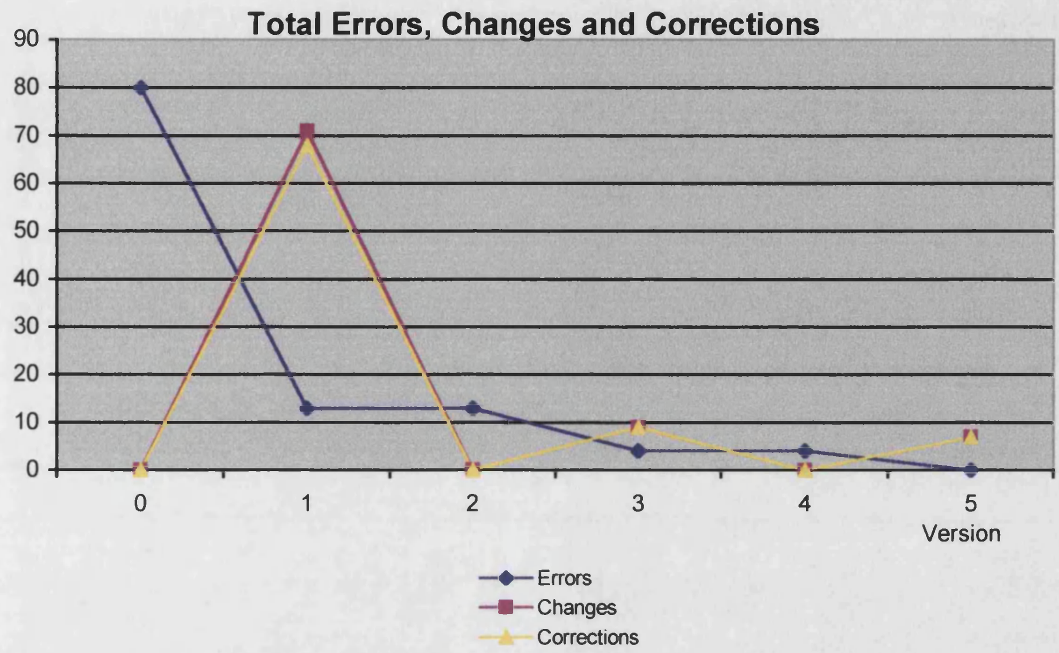


Figure 5.25 Results for model 12 (a)



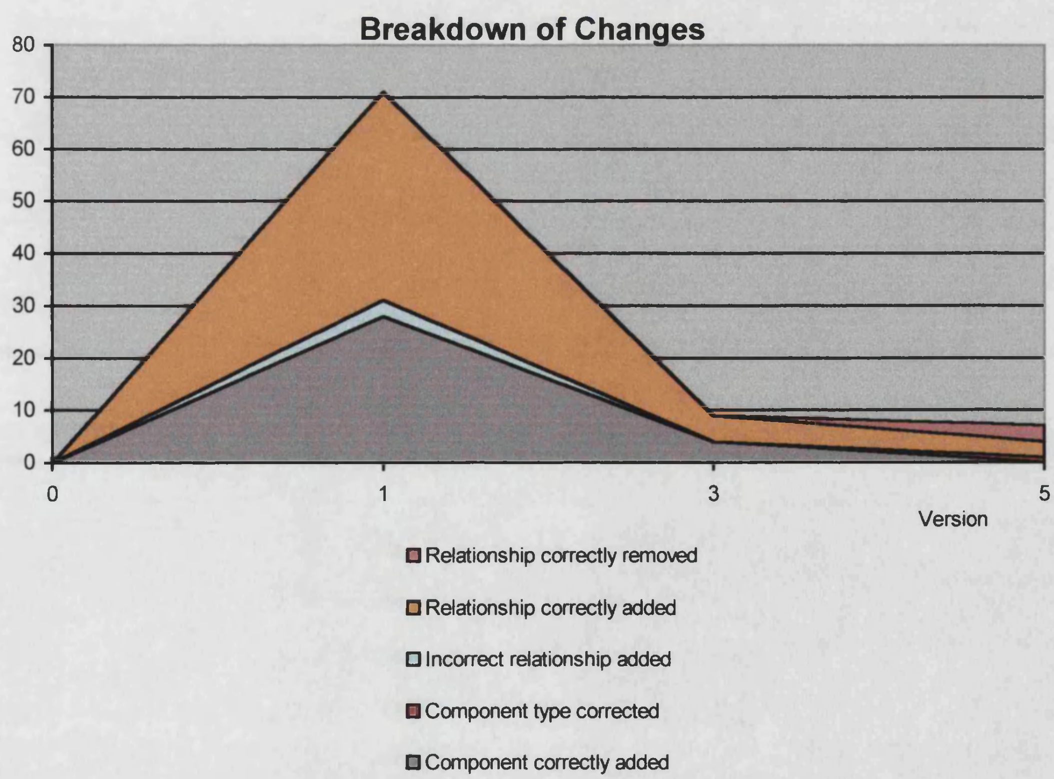


Figure 5.26 Results for model 12 (b)

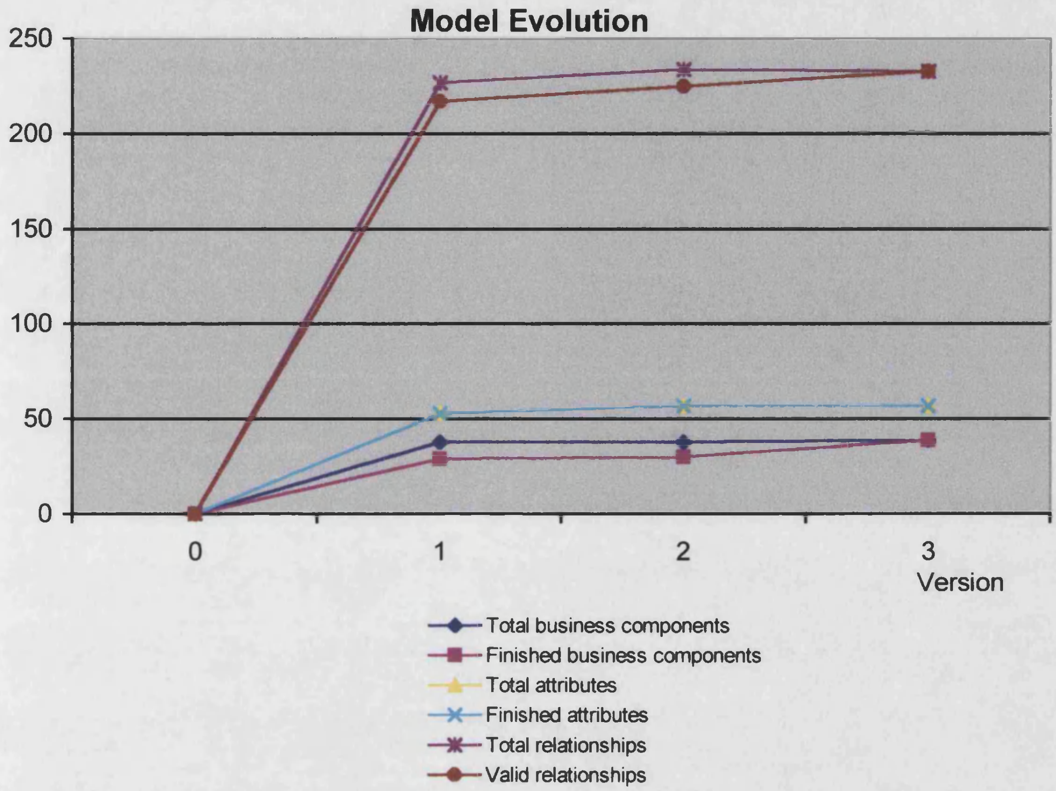
13. Operators and Networks Model

This large model was produced by the same modeller as the previous four models. Although the model was almost complete in its first version, about 5% of relationships remained incorrect until the final version. The modeller's productivity was once again not as good as that for models 9 and 10, but the error rate was low. Again the quality of the model was judged to be sufficient that no corrected version was considered necessary.

No of components	96	Changes per component	3.23
No of relationships	233	Errors per change	0.03
Relationships per component	5.97	Components per hour	1.95
Attributes per component	1.46	No. of changes	312

Breakdown of changes	Total	%
Relationship correctly added	194	62.17
Component correctly added	96	30.76
Incorrect relationship added	11	3.52
Relationship correctly removed	11	3.52

Table 5.13 Results for model 13



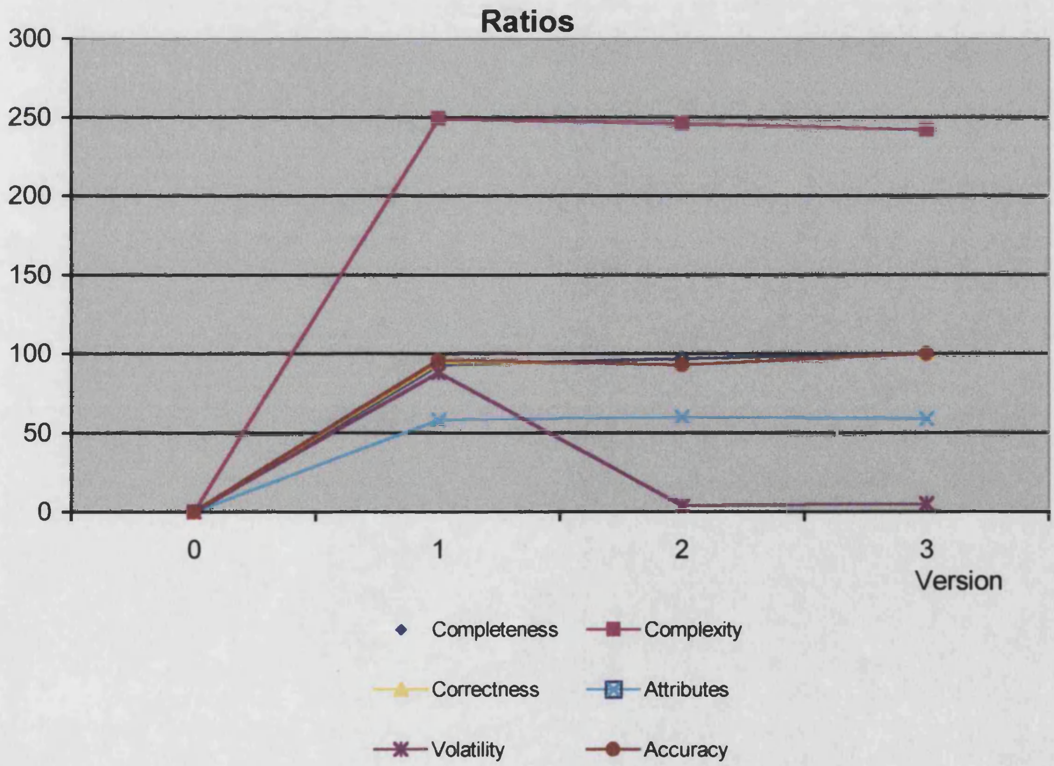
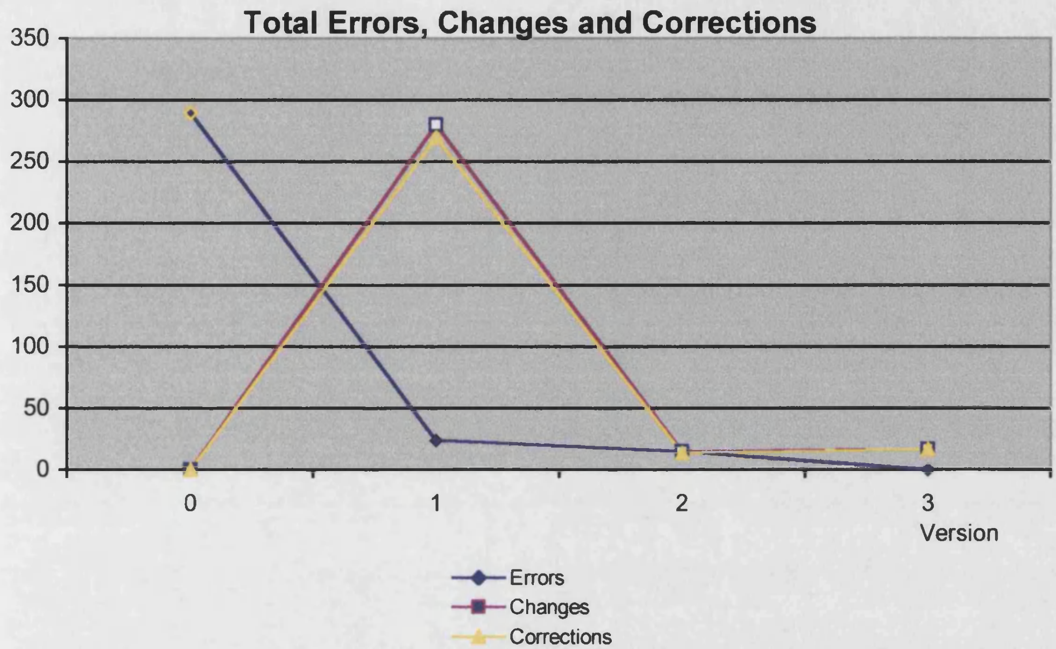


Figure 5.27 Results for model 13 (a)



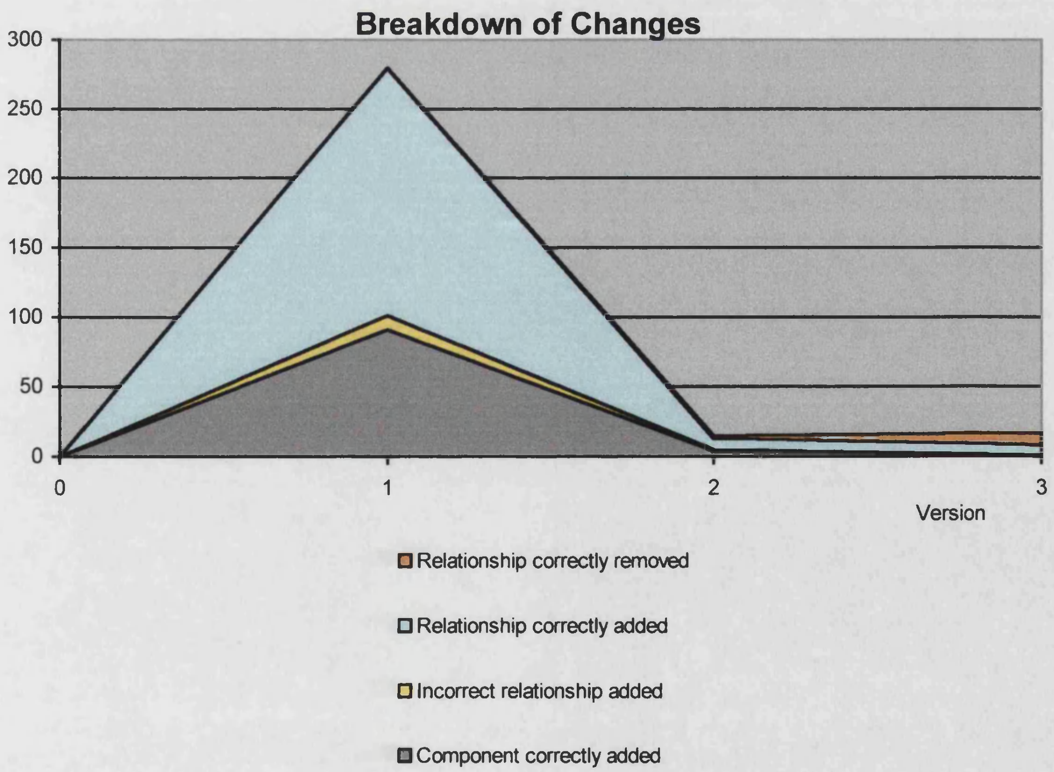


Figure 5.28 Results for model 13 (b)

14. Purchase Order Processing Model

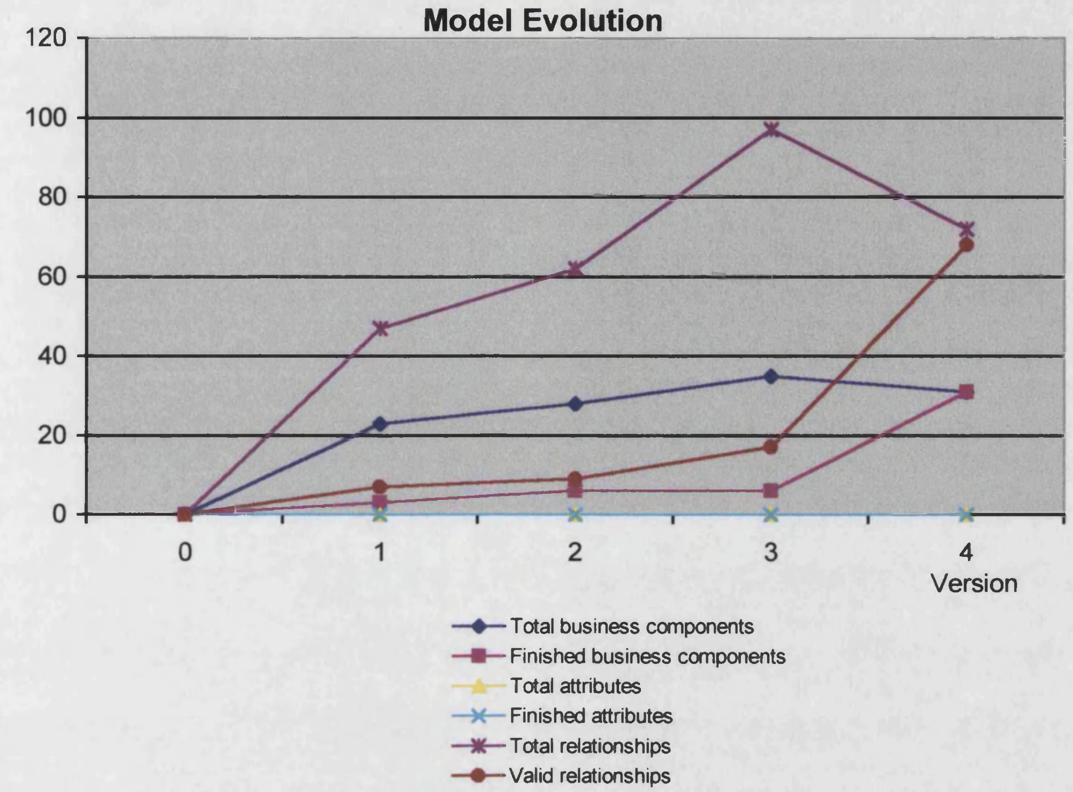
This model was produced by a trainee systems analyst using the same methods as for model 3. The model is not a particularly large one but, as in model 3, both the number of errors and the level of complexity increased rapidly and failed to decline. The high number of errors was due mainly to the presence of many of incorrect relationships. In the finished (uncorrected) model (version 3) there were nearly 50% too many relationships overall, and only about 20% of the relationships were correct. Consequently, the error rate was high. As before, the modeller remained unaware of problems in the model and was satisfied that her end result (version 3) was correct. The corrected version of the model (version 4) used almost the same components but showed a significant reduction in overall complexity. The modeller made the same mistakes as in model 3, plus some additional errors.

- Incorrect cardinalities were included (for example, the model allows a purchase order to have no purchase order lines).
- Some key concepts were treated as attributes or omitted (for example, 'Manufacturer code' is an attribute of 'Product' but the concept 'Manufacturer') is absent).

No of components	31	Changes per component	7.24
No of relationships	72	Errors per change	0.56
Relationships per component	2.32	No. of changes	225
Attributes per component	0.00		

Breakdown of changes	Total	%
Relationship correctly removed	60	26.66
Incorrect relationship added	53	23.55
Relationship correctly added	33	14.66
Component correctly added	31	13.77
Incorrect component added	21	9.33
Component correctly removed	21	9.33
Cardinality corrected	4	1.77
Cardinality altered wrongly	2	0.88

Table 5.14 Results for model 14



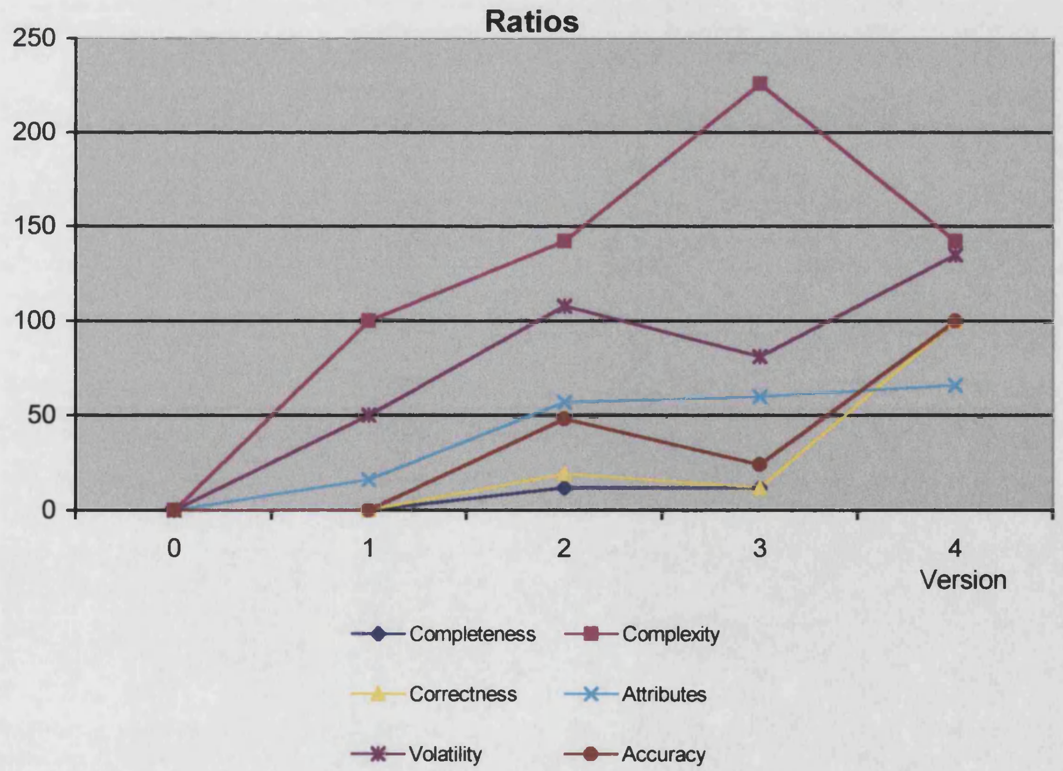
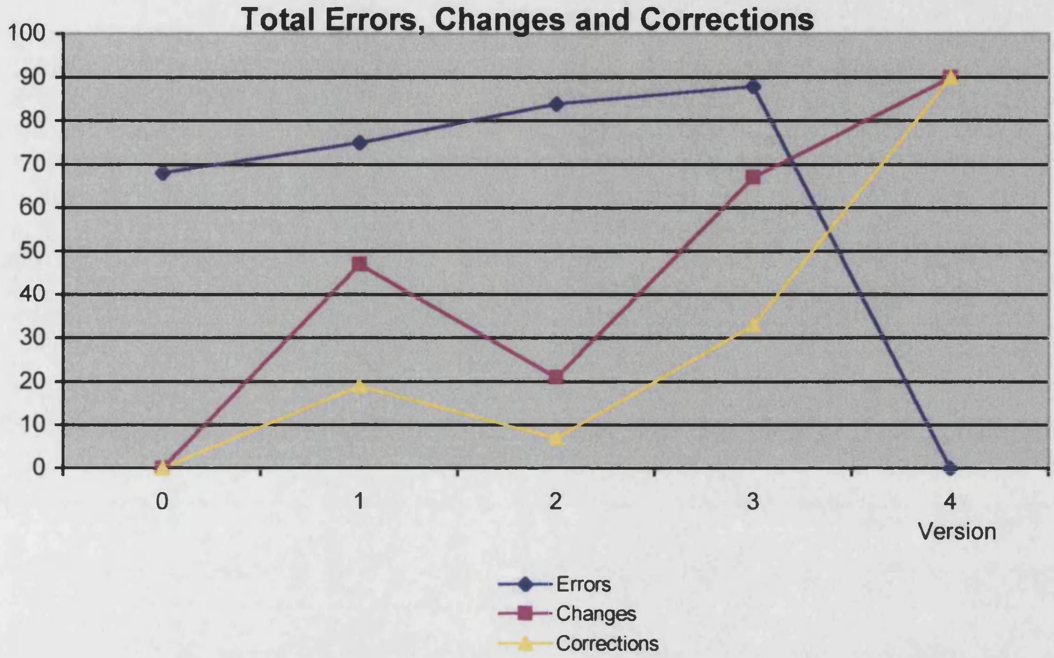


Figure 5.29 Results for model 14 (a)



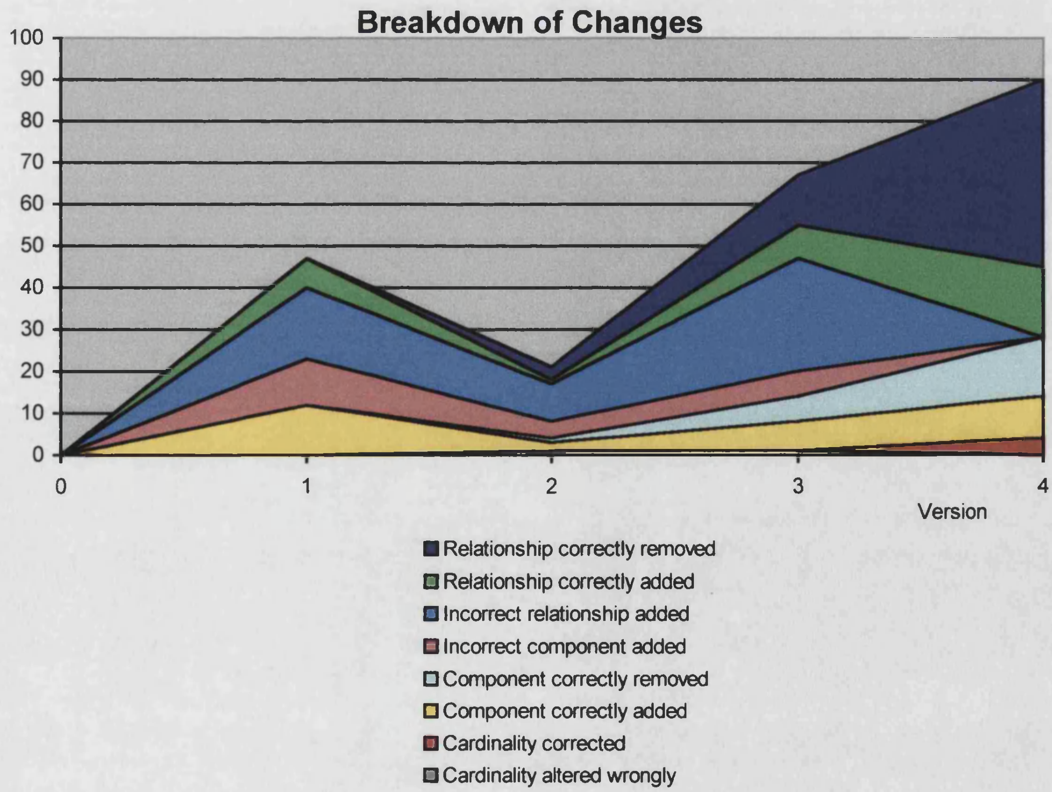


Figure 5.30 Results for model 14 (b)

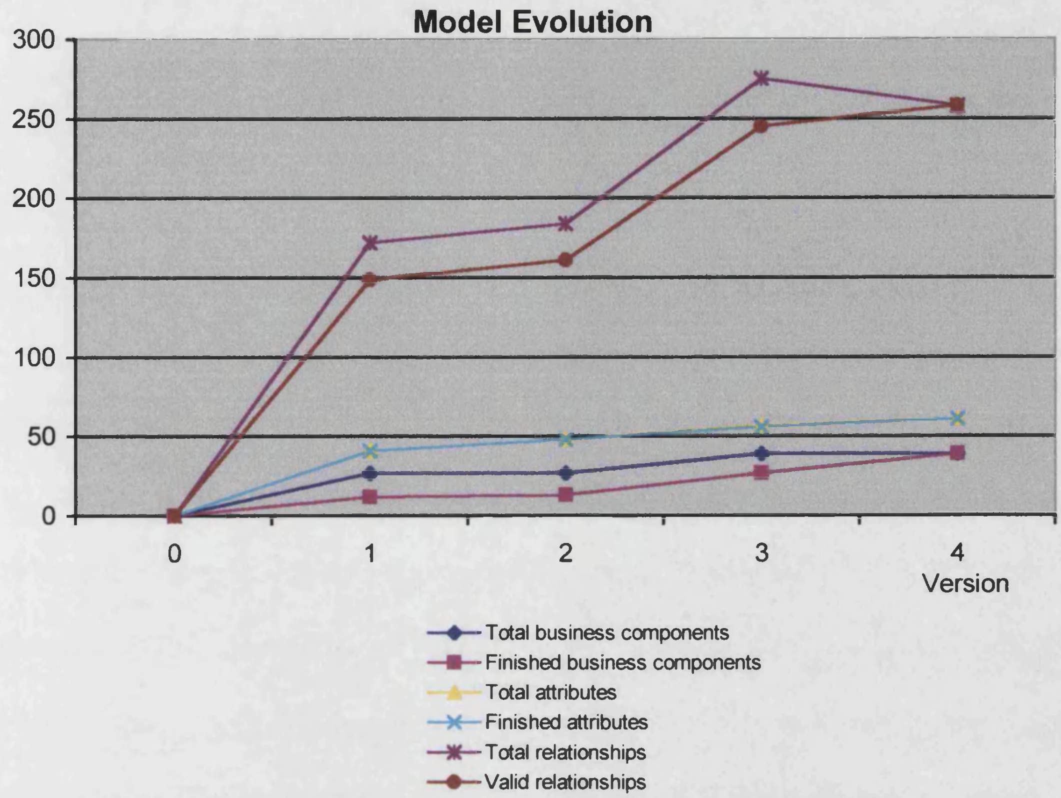
15. Secretariat Model

This model was produced by the experienced analyst, and is a very large model with high complexity. In contrast with other models produced by the same modeller, the initial version of this model was only 60% complete and the error rate relatively high (although not comparable to the error rates produced by inexperienced modellers). The number of relationships increased by approximately two-thirds after the first version. As in other models, a consistent number (approximately 15% in this case) of incorrect relationships persisted until they were corrected in the final model. However, accuracy tended towards 100%. The quality of the model was judged sufficient that no corrected version was considered necessary.

No of components	100	Changes per component	4.01
No of relationships	258	Errors per change	0.10
Relationships per component	6.61	Components per hour	1.30
Attributes per component	1.56	No. of changes	401

Breakdown of changes	Total	%
Relationship correctly added	220	54.86
Component correctly added	100	24.93
Relationship correctly removed	38	9.47
Incorrect relationship added	38	9.47
Incorrect component added	2	0.49
Component correctly removed	2	0.49
Relationship wrongly removed	1	0.24

Table 5.15 Results for model 15



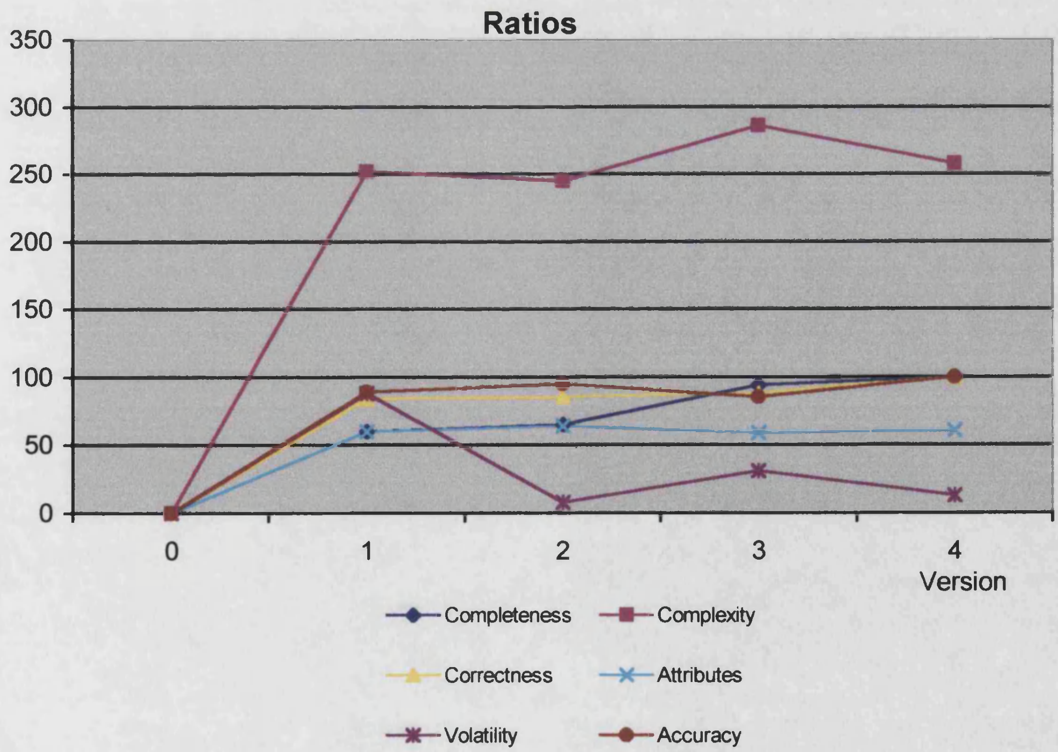
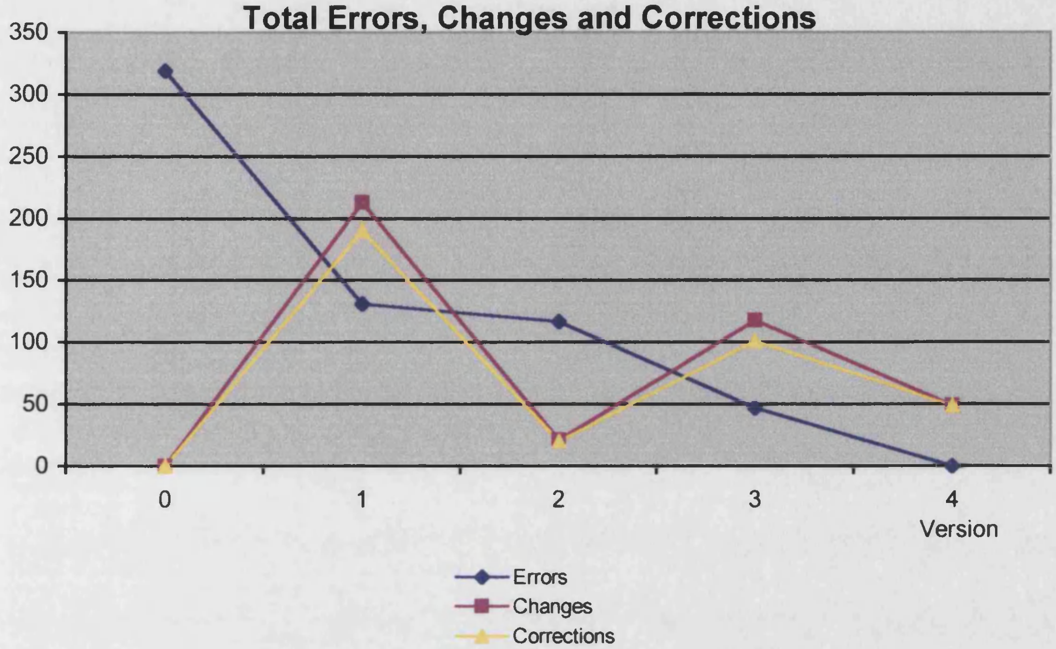


Figure 5.31 Results for model 15 (a)



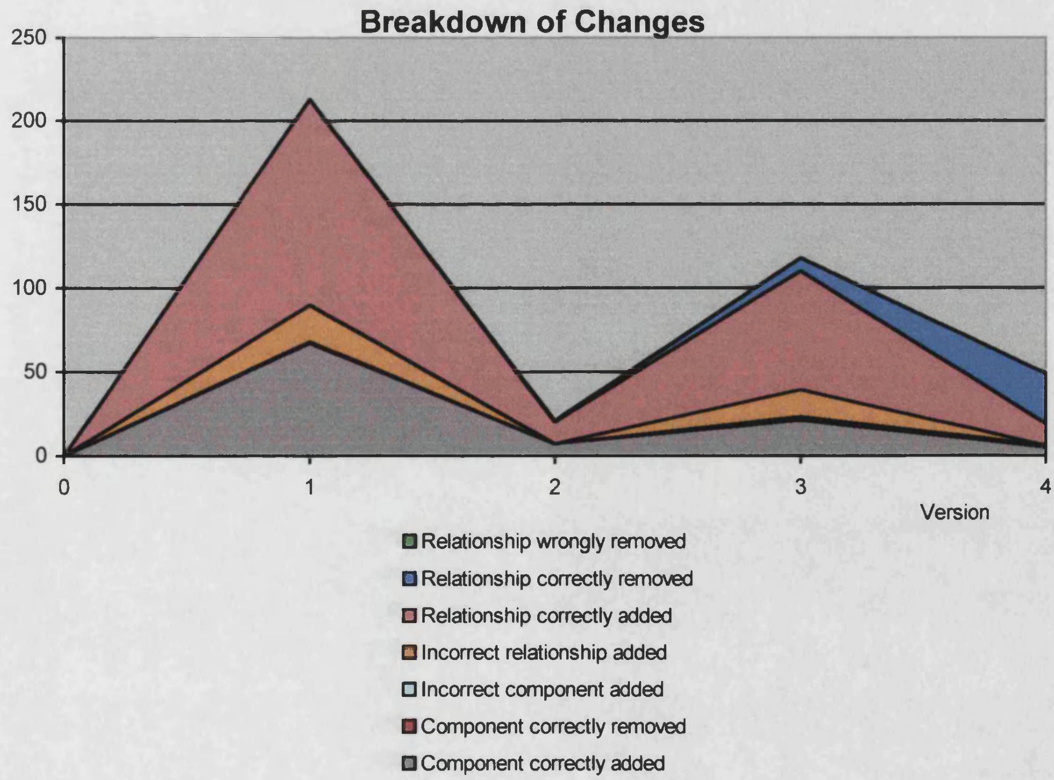


Figure 5.32 Results for model 15 (b)

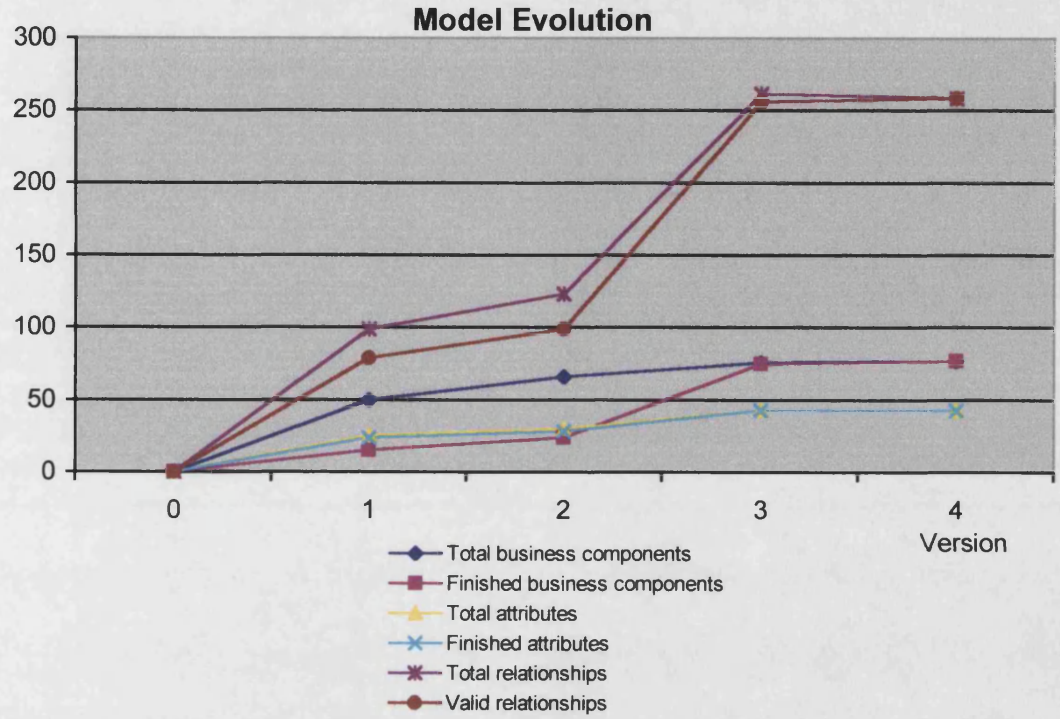
16. Security Group Model

This very large, highly complex model was produced by the same experienced analyst as the previous model. Unlike smaller models, this one was only 40% complete after the first version but, by version 3, was almost 100% complete. The modeller made relatively few errors (e.g. only 3 cardinality corrections out of 500 changes). Accuracy of changes improved as modelling progressed, quickly reaching a high value and then tending towards 100%. The modeller's productivity was high.

No of components	120	Changes per component	3.33
No of relationships	259	Errors per change	0.11
Relationships per component	3.36	Components per hour	3.85
Attributes per component	0.55	No. of changes	493

Breakdown of changes	Total	%
Relationship correctly added	231	46.85
Component correctly added	147	29.81
Relationship correctly removed	30	6.08
Incorrect relationship added	30	6.08
Incorrect component added	25	5.07
Component correctly removed	25	5.07
Cardinality corrected	3	0.60
Component wrongly removed	1	0.20
Component type corrected	1	0.20

Table 5.16 Results for model 16



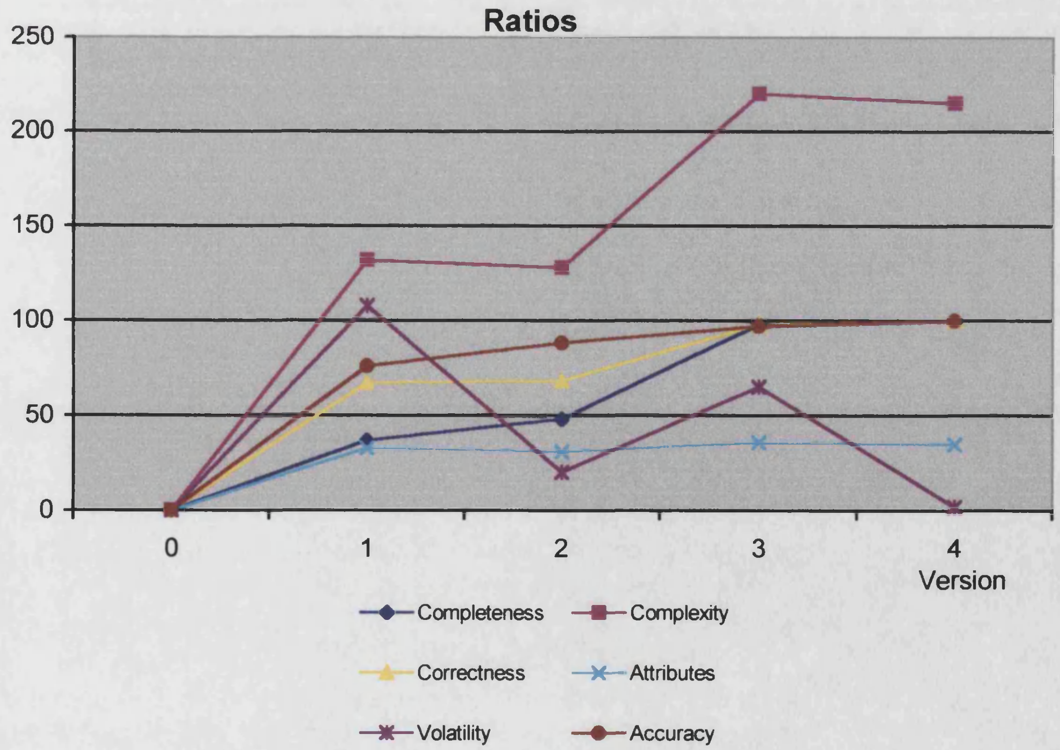
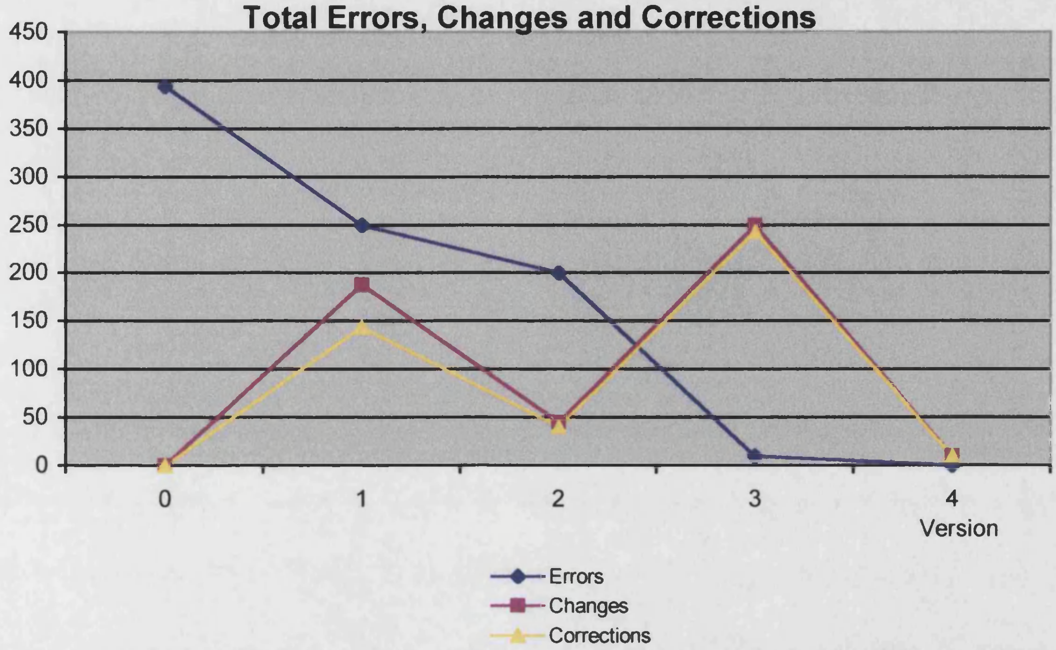


Figure 5.33 Results for model 16 (a)



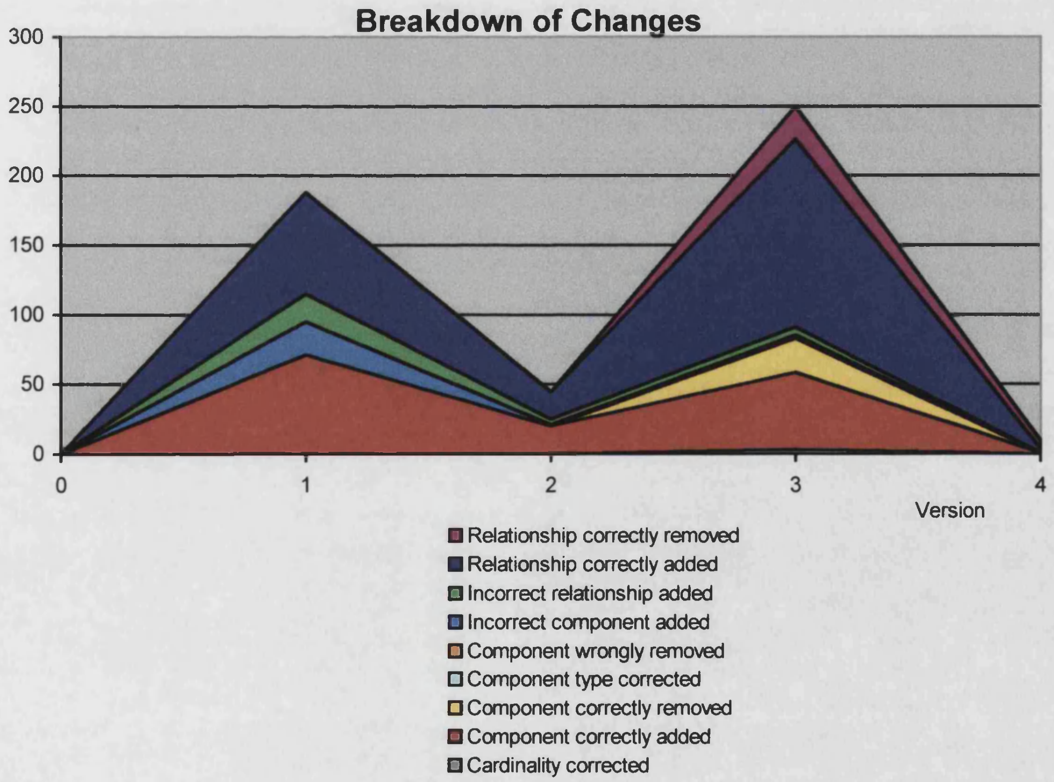


Figure 5.34 Results for model 16 (b)

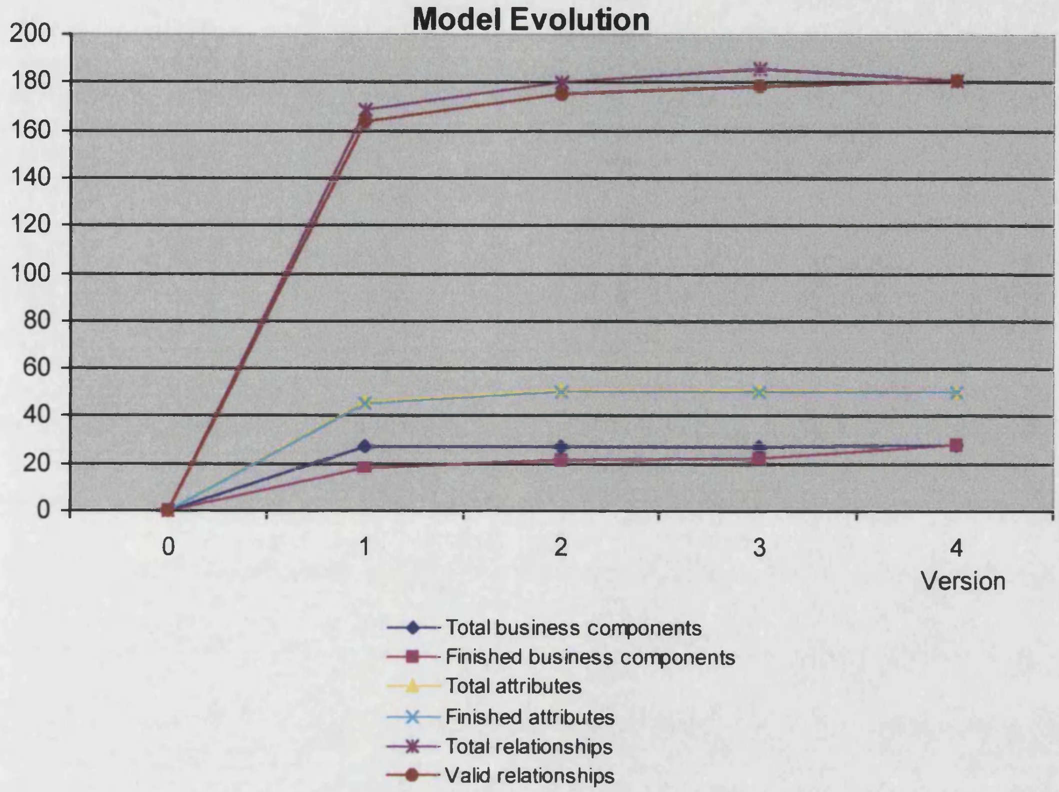
17. Security/Fraud Model

This model was produced by the same analyst as the previous two models. The model is a relatively large and complex one. It was 91% complete in the first version and 97% complete by the second version. A small percentage (<5%) of relationships remained incorrect until the last version. Productivity and error rate were similar to the other models produced by the same modeller using method 'Y'. The quality of the model was judged sufficient that no corrected version was considered necessary.

No of components	78	Changes per component	3.07
No of relationships	181	Errors per change	0.02
Relationships per component	6.46	Components per hour	1.40
Attributes per component	1.78	No. of changes	246

Breakdown of changes	Total	%
Relationship correctly added	153	62.19
Component correctly added	78	31.70
Incorrect relationship added	7	2.84
Relationship correctly removed	7	2.84
Component type corrected	1	0.40

Table 5.17 Results for model 17



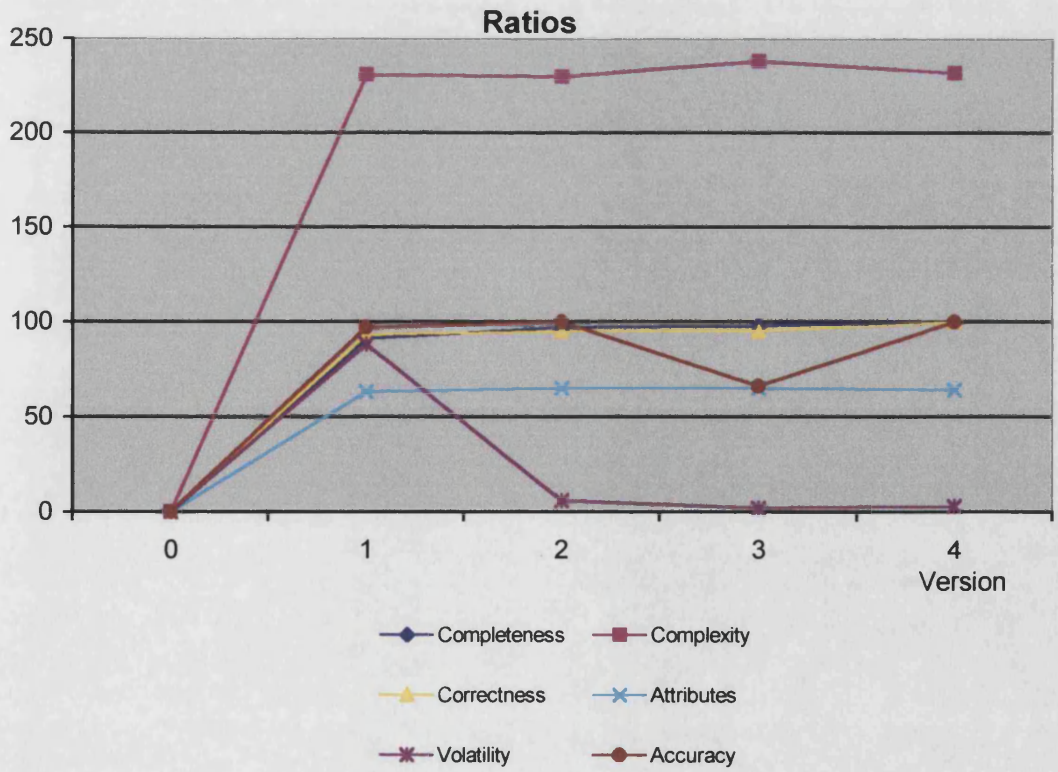


Figure 5.35 Results for model 17 (a)

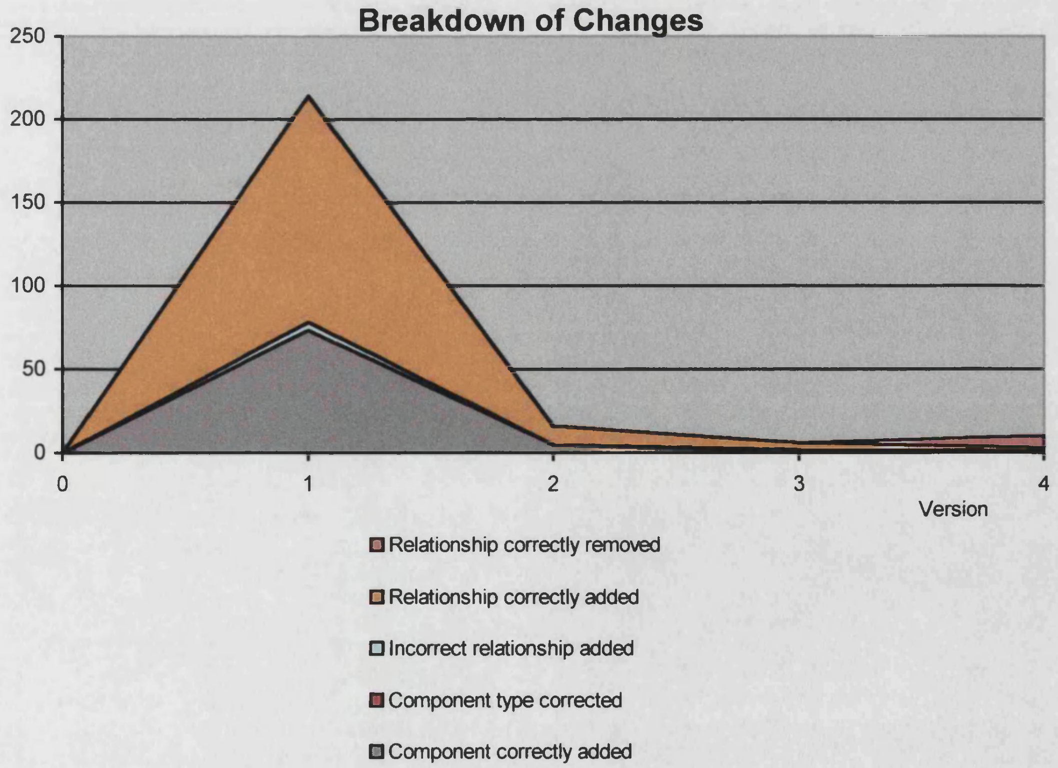


Figure 5.36 Results for model 17 (b)

18. Stock Control Model

This model was produced by a company director and lecturer with many years' business experience. The modeller had only limited experience in the use of computers and no knowledge of database technology or database design. In this model the modeller started poorly by creating a number of incorrect components. The number of errors increased sharply with successive versions until the model was large, highly over-complex and confused. Overall, more than two-thirds of all changes made to the model were incorrect. The modeller expressed satisfaction with the end result (version 3) which was, in fact, very poorly structured. Almost all of the relationships (approximately 90%) in this version of the model were incorrect. This modeller was given more help than other modellers since it became apparent that the model was unlikely to get off the ground otherwise. The modeller appeared to grasp the modelling notation relatively quickly but, ultimately, did not structure the model logically. Large numbers of incorrect relationships were inserted between versions 2 and 3. The corrected version of the model (version 4) contained roughly the same concepts as earlier versions but related them quite differently. Overall, the modeller took an entirely impressionistic approach, and did not pay serious attention to correcting the formal meaning of the model. The productivity figure is very low chiefly because none of the business components in the model were actually finished (according to the definition given in Chapter 5).

No of components	42	Changes per component	6.07
No of relationships	60	Errors per change	0.68
Relationships per component	4.28	Components per hour	0.00
Attributes per component	2.00	No. of changes	261

Table 5.18 Results for model 18 (a)

Breakdown of changes	Total	%
Relationship correctly added	58	22.22
Relationship correctly removed	47	18.00
Incorrect relationship added	47	18.00
Component correctly added	45	17.24
Incorrect component added	27	10.34
Component correctly removed	27	10.34
Relationship wrongly removed	4	1.53
Component wrongly removed	3	1.14
Cardinality altered wrongly	2	0.76
Component type changed wrongly	1	0.38

Table 5.18 Results for model 18 (b)

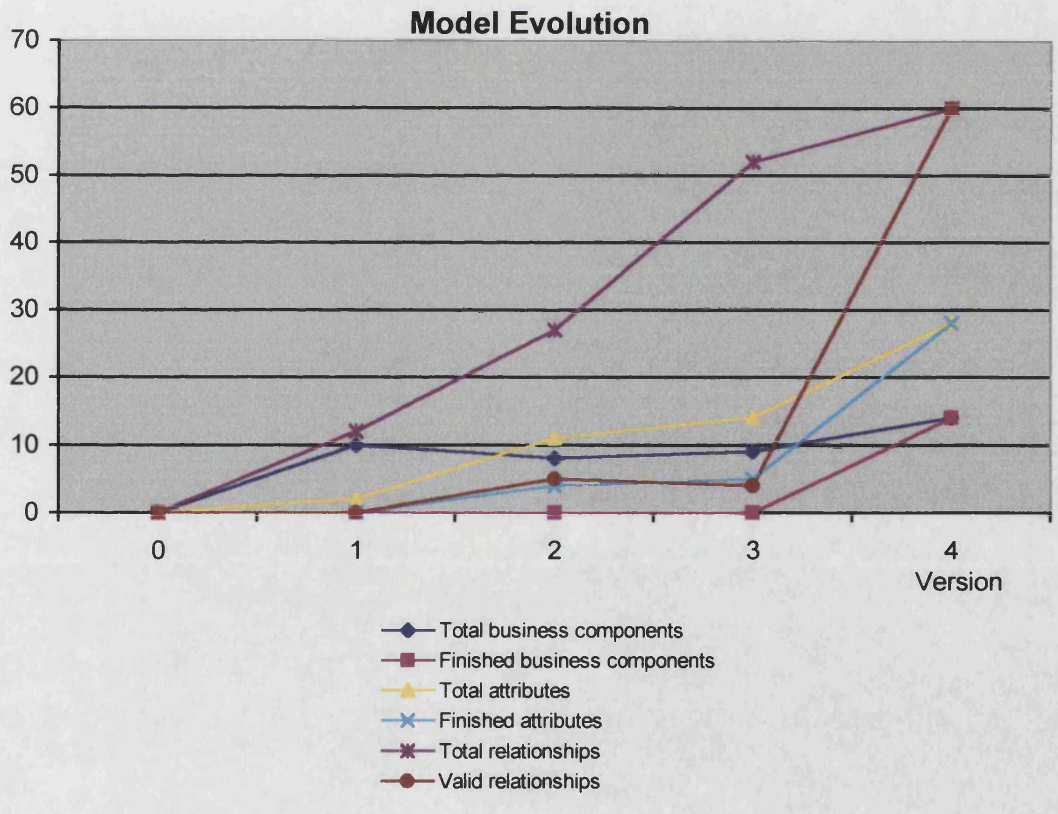
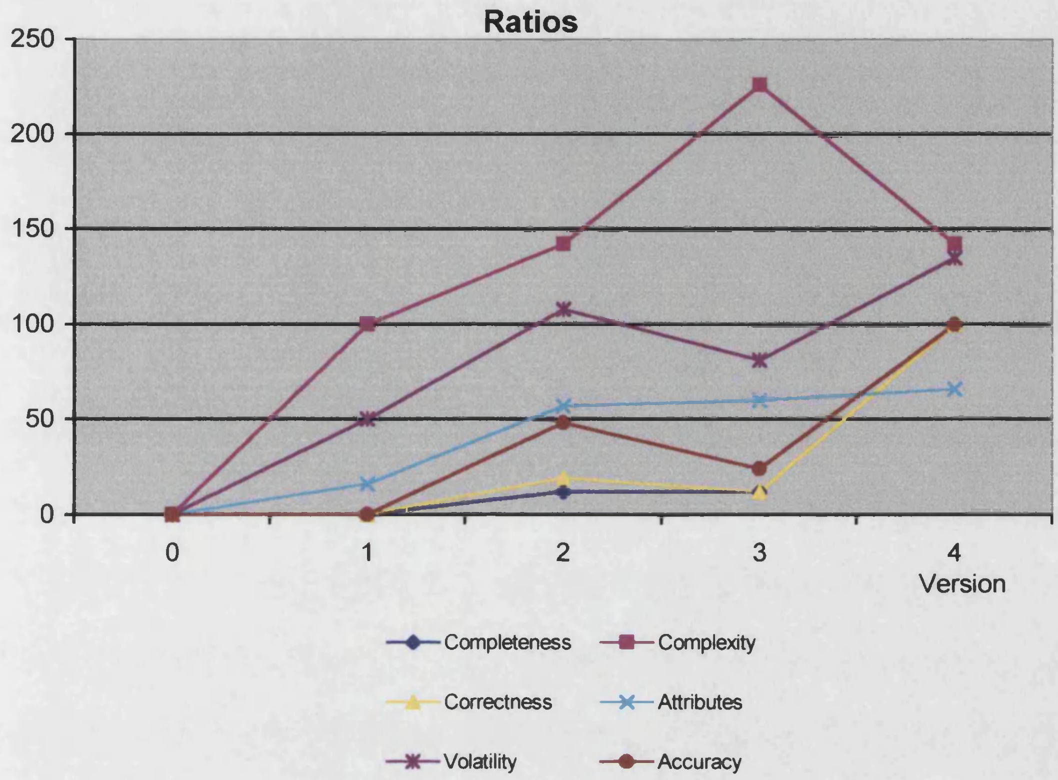


Figure 5.37 Results for model 18 (a)



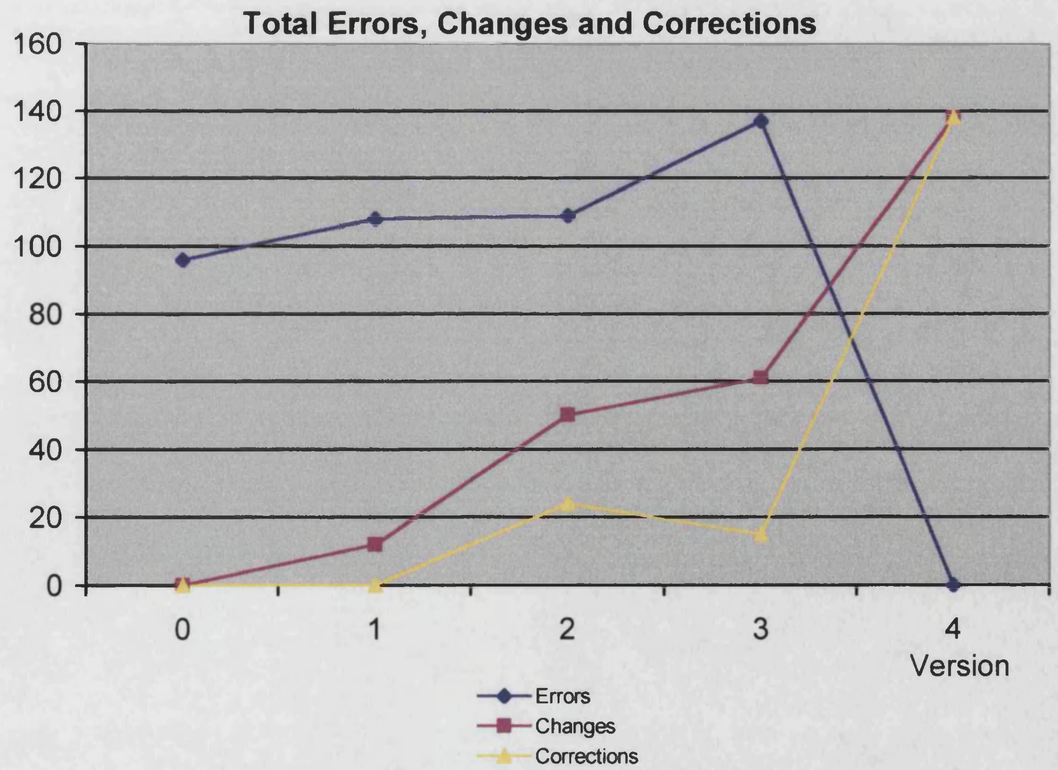


Figure 5.38 Results for model 18 (b)

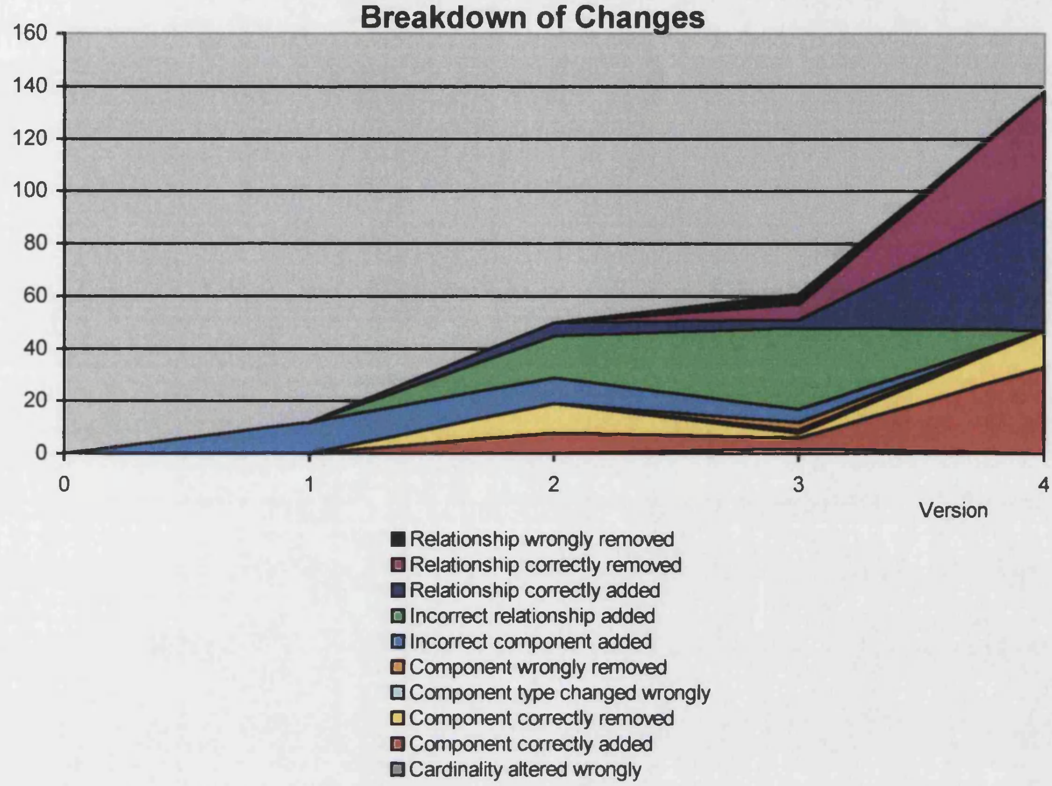


Figure 5.39 Results for model 18 (c)

19. Theatrical Productions Model

This medium-sized model was produced by an audio-visual technician with experience of theatrical production management. The modeller's use of computers had been limited mainly to digital video editing equipment and he had no knowledge of system development techniques such as object or data modelling. During the development of the model its completeness and correctness increased smoothly, and the modeller's accuracy improved throughout. The error count initially declined, increased slightly and then declined again as the modeller experimented with ways of representing the subject matter. The overall error rate was intermediate but productivity was high. A corrected version of this model was produced (version 17).

No of components	39	Changes per component	4.14
No of relationships	54	Errors per change	0.28
Relationships per component	2.70	Components per hour	3.66
Attributes per component	0.95	No. of changes	183

Table 5.19 Results for model 19 (a)

Breakdown of changes	Total	%
Relationship correctly added	53	28.96
Component correctly added	39	21.31
Relationship correctly removed	31	16.93
Incorrect relationship added	31	16.93
Cardinality corrected	9	4.91
Incorrect component added	7	3.82
Component correctly removed	7	3.82
Cardinality altered wrongly	3	1.63
Relationship wrongly removed	2	1.09
Component type corrected	1	0.54

Table 5.19 Results for model 19 (b)

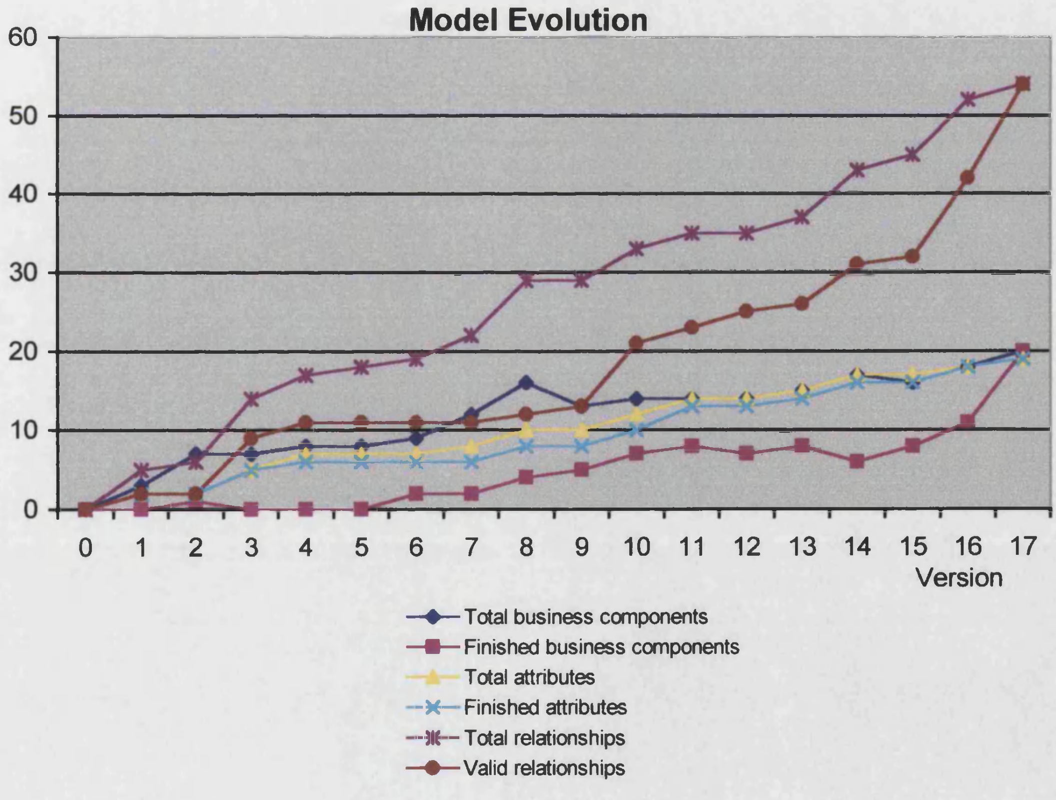
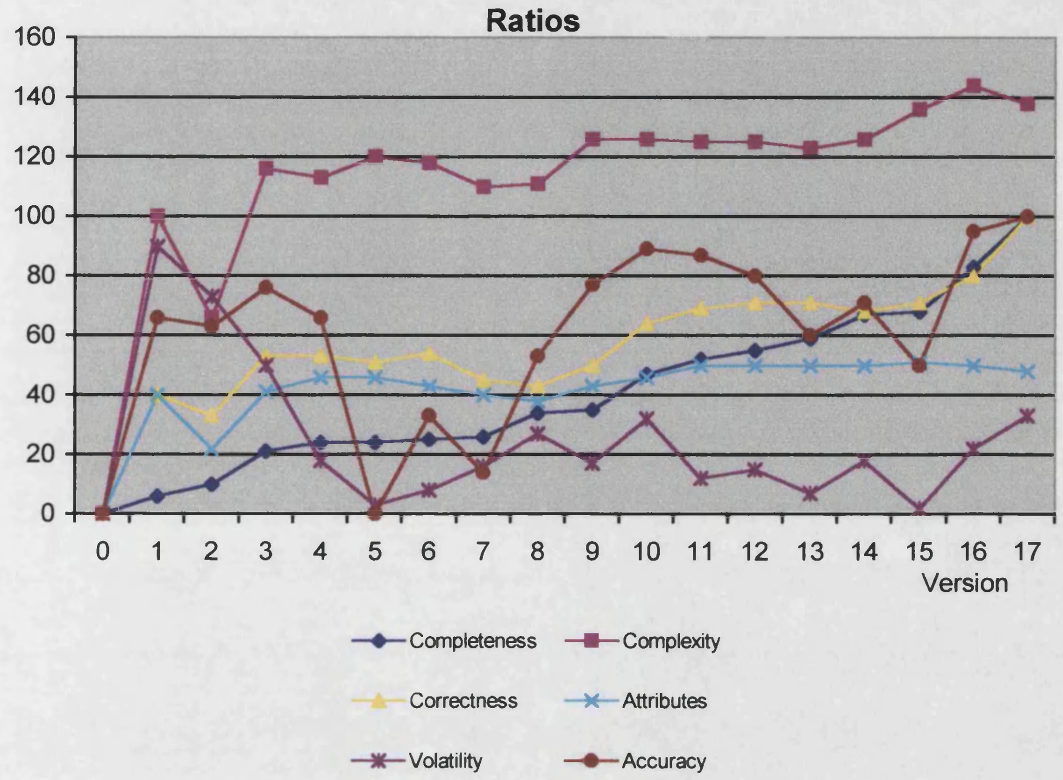


Figure 5.40 Results for model 19 (a)



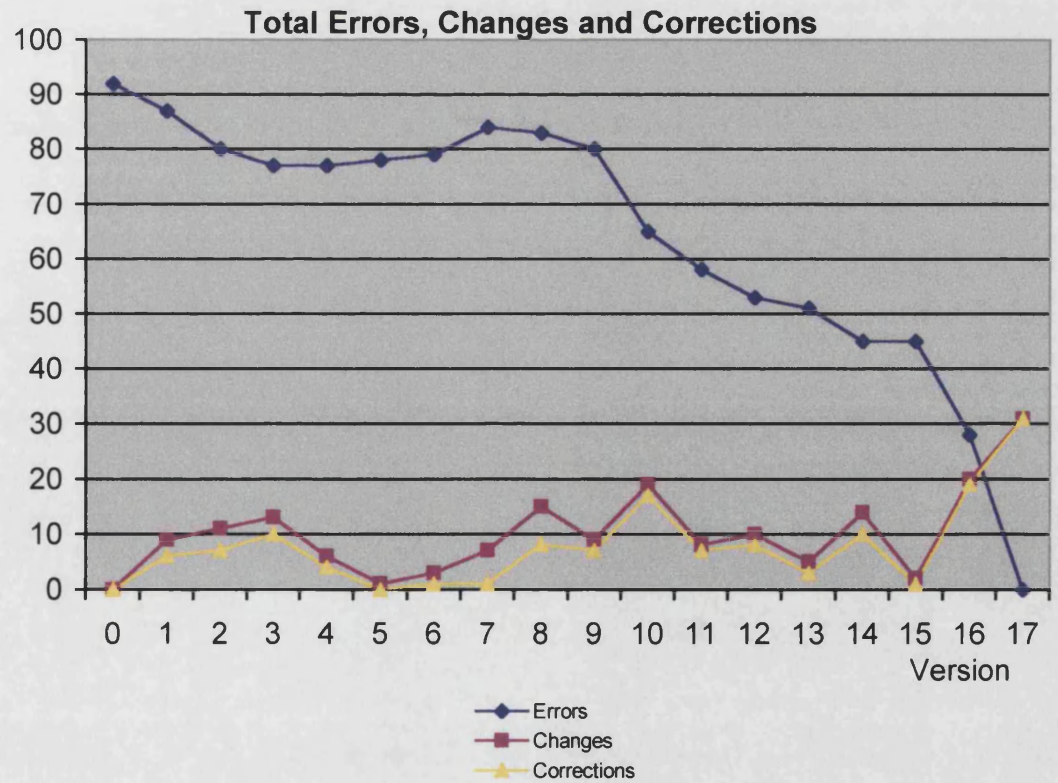


Figure 5.41 Results for model 19 (b)

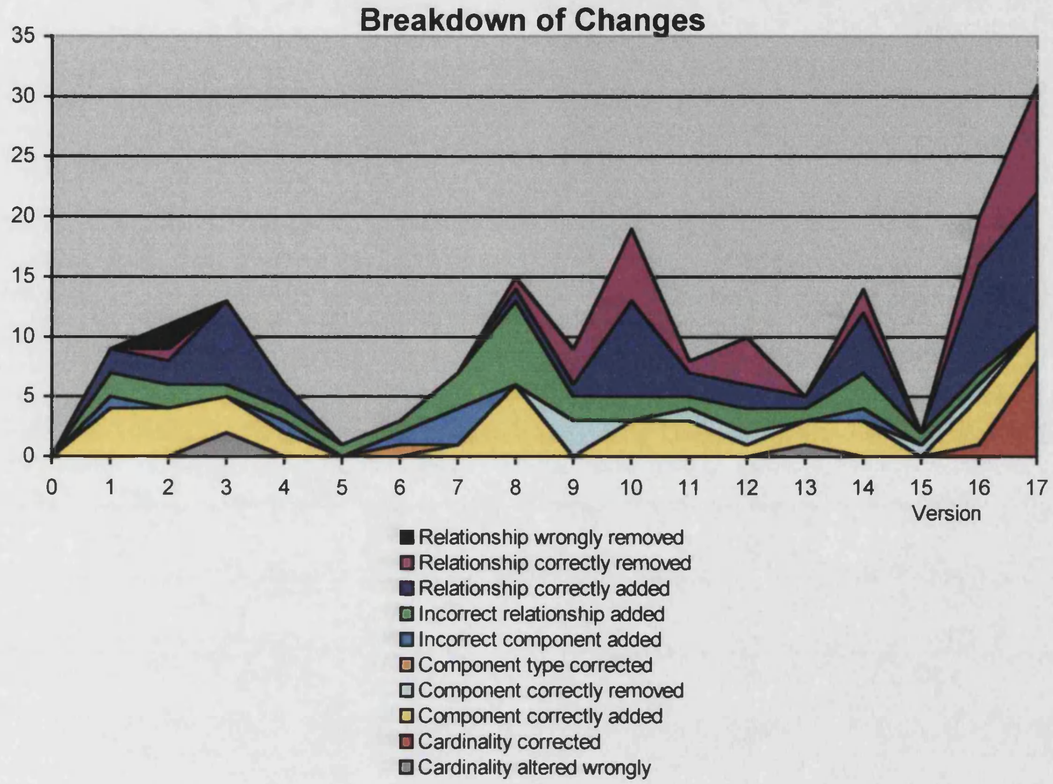


Figure 5.42 Results for model 19 (c)

Summary

The tables below summarise the results for each model.

Measure	Model (experienced modeller only)							
	9	**10**	**11**	**12**	**13**	**15**	**16**	**17**
Method	X	X	Y	Y	Y	Y	X	Y
Modeller's experience level	High	High	High	High	High	High	High	High
No. components	60	65	45	32	96	100	120	78
No. relationships	110	93	67	59	233	258	259	181
Changes	217	201	107	87	312	401	493	246
Relationships/concept	2.34	2.06	6.09	5.9	5.97	6.61	3.36	6.46
Attributes/concept	.27	0.44	3.09	2.2	1.46	1.56	.55	1.78
Changes per concept	2.72	2.52	2.37	2.49	3.23	4.01	3.33	3.07
Correctness (%)	100	100	100	100	100	100	100	100
Complexity	183	143	148	184	242	258	215	232
Completeness (%)	100	100	100	100	100	100	100	100
Error rate	0.07	0.08	0.02	0.03	0.03	0.1	0.11	0.02
Productivity	3.91	4.5	1.1	2.5	1.95	1.3	3.85	1.4
Effectiveness	181.81	207	53.9	121.25	94.57	58.5	171.32	68.6

Table 5.20 Summary of results for main study (experienced modeller)

Measure	Model (inexperienced modellers only)									
	1	2	3	5	6	7	8	14	18	19
Method	X	Y	Y	X	X	X	Y	Y	Y	X
Experience level	Some	Some	Some	None	Some	None	None	Some	None	None
No. components	38	50	29	99	28	20	79	31	42	39
No. relationships	44	69	63	171	40	22	96	72	60	54
Changes	195	424	140	1379	194	55	619	225	261	183
Relationships/concept	4.4	4.05	2.17	3.8	4.44	1.57	4.36	2.32	4.28	2.7
Attributes/concept	2.8	1.94	0	1.2	2.11	.42	2.59	0	2	0.95
Changes per concept	4.41	8.04	4.81	10.63	5.57	2.75	7.59	7.24	6.07	4.14
Correctness (%)	100	41	18	68	81	85	23	17	12	80
Complexity	115	234	246	200	150	110	172	277	226	144
Completeness (%)	100	65	30	81	85	80	31	36	12	83
Error rate	0.28	0.51	0.59	0.42	0.34	0.14	0.63	0.56	0.68	0.28
Productivity	4	1.33	N/A	N/A	2.4	4.33	1	N/A	0	3.66
Effectiveness	144	8.68	N/A	N/A	54.52	126.6	1.31	N/A	0	87.48

Table 5.21 Summary of results for main study (inexperienced modeller)

Average values for all measures are summarised in Table 5.22.

		Experienced modeller		Inexperienced modeller	
Measure	**Source:**	**Method 'Y'**	**Method 'X'**	**Method 'Y'**	**Method 'X'**
Volatility	Average	3.03	2.46	6.75	5.50
	Standard deviation	*0.65*	*0.86*	*1.30*	*3.03*
Completeness	Average	100.00	100.00	34.79	85.80
	Standard deviation	*0.00*	*0.00*	*19.17*	*8.16*
Correctness	Average	100.00	100.00	22.20	82.80
	Standard deviation	*0.00*	*0.00*	*11.21*	*11.51*
Productivity	Average	1.65	4.08	0.77	3.59
	Standard deviation	*0.57*	*1.18*	*0.69*	*0.84*
Error rate	Average	0.04	0.06	0.59	0.29
	Standard deviation	*0.03*	*0.04*	*0.06*	*0.10*
Effectiveness	Average	79.36	171.60	3.33	103.15
	Standard deviation	*28.21*	*50.62*	*4.67*	*40.11*
Model size		70.20	81.70	46.20	44.80
Changes		261.5	254.50	34.8	85.8

Table 5.22 Average measures for different modellers using each technique

Table 5.23 gives the differences in volatility, model size and productivity measured between the two modelling techniques, for experienced and inexperienced modellers.

Measure	Δ (experienced modeller)	Δ (inexperienced modeller)
Volatility	-18.81%	-18.52%
Model size	+11.64%	-3.03%
Productivity	+147.27%	+366.23%

Table 5.23 Differences between results for method 'Y' and method 'X' (Δ = X - Y)

Changes

Table 5.24 and Figure 5.43 summarise the relative frequencies of different types of change made to models, for the two techniques. As may be seen, there is very little substantive difference in the profile of changes between the two techniques (correlation .995).

Change	Method 'Y'	Method 'X'
Relationship correctly added	31.00	29.05
Component correctly added	22.10	19.42
Relationship correctly removed	17.87	17.86
Incorrect relationship added	17.31	17.80
Component correctly removed	4.86	6.31
Incorrect component added	4.86	6.31
Relationship wrongly removed	0.78	0.61
Cardinality altered wrongly	0.48	0.42
Cardinality corrected	0.37	1.62
Component wrongly removed	0.33	0.55

Table 5.24 Change frequencies

For inexperienced modellers, using both techniques, creation of incorrect relationships outweighs creation of correct relationships. For experienced modellers the situation is reversed, with creation of correct relationships dominating.

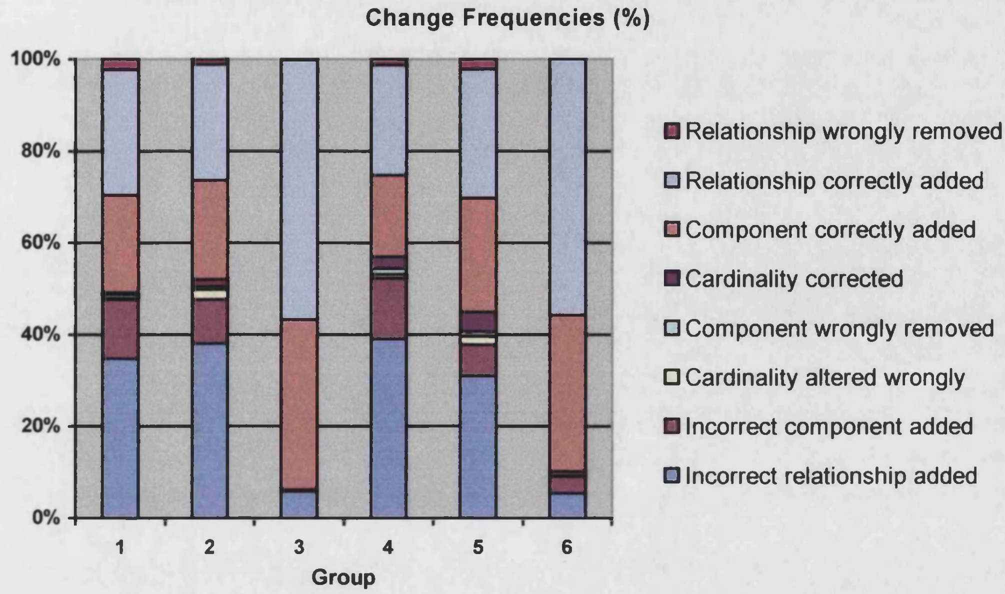


Figure 5.43 Change frequencies

5.2 Secondary study

According to the results of the secondary study, on average 58% of the relevant business concepts were correctly identified. However, only approximately 30% of relationships were correctly identified, and this reduces to 25% of relationships if cardinality is taken into account. Most of the relationships included by the modellers in their models were therefore incorrect ones. The average 'completeness' figure for the models (according to the definition used in Chapter 4) was 43.27%. Table 5.25 summarises the results of the secondary study.

5.3 Questionnaire responses

The questionnaires were completed for group modelling sessions conducted by the experienced analyst (modeller J). A total of 15 questionnaires were returned by modelling session participants.

Ratings by participants

On completion of each modelling session the participants were asked to rate several factors on a scale of 1-5 (low to high). Each participant was also asked to state their own education level and whether or not they had created a database at any time in the past. This allowed individuals to be grouped according to their characteristics so that average ratings could be calculated.

Measure	Model			Average
	Retail banking	Health spa	Mobile phones	
Sample size (no. of models analysed)	5	26	15	
Classes				
Expected number	12	17	9	
Average number (% of expected number)	88.33	75.11	97.77	87.07
Average % of expected business concepts present	73.33	41.17	59.25	57.92
Average % well-named	66.66	33.48	48.88	49.67
Average % named incorrectly as plural	0.00	5.65	0.74	2.13
Associations				
Expected number	12	23	9	
Average number (as % of expected number)	104.35	60.62	94.67	86.55
Average % between correct classes	54.99	14.71	20.13	29.94
Average % with correct cardinality	48.33	11.03	16.66	25.34

Table 5.25 Results of secondary study

Question	Response
How valuable do you think the information modelling sessions have been?	80% rated the value of the sessions at 3 ("Reasonably valuable") or higher. 20% rated the value of the sessions at 5 ("Very valuable").
How well do you understand the modelling technique?	All rated their understanding at 3 ("Reasonably well") or higher. 13% rated their understanding at 5 ("Very well").
How well do you understand the modelling software?	86% rated their understanding at 3 ("Reasonably well") or higher. 13% rated their understanding at 5 ("Very well").
How complete is the model now?	All rated the model completeness at 3 ("Reasonably complete") or higher. 33% rated the model completeness at 4 ("Nearly complete"). None rated the model completeness at 5 ("100% complete").
How correct is the model now?	93% rated the model correctness at 3 ("Reasonably complete") or higher. 13% rated the model correctness at 5 ("100% correct").

Table 5.26 Summary of questionnaire responses by modelling session participants

75% of the participants claimed to be educated to diploma or degree level and 50% said they had created a database at some point in the past (of which all had used either Microsoft Access or Lotus Notes).

	Diploma/degree?		Created DB before?		Modelling session	
	No	Yes	Yes	No	Initial	Follow-up
% of participants	46%	53%	46%	53%	80%	20%
Valuable	3.42	4.00	4.14	3.37	3.66	4.00
Standard deviation	*1.13*	*0.92*	*0.37*	*1.30*	*1.15*	*0.00*
Understand technique	3.85	3.50	4.14	3.25	3.66	3.66
Standard deviation	*0.89*	*0.53*	*0.69*	*0.46*	*0.77*	*0.57*
Completeness	2.85	3.37	3.57	2.75	3.00	3.66
Standard deviation	*1.34*	*0.51*	*0.53*	*1.16*	*1.04*	*0.57*
Correctness	3.42	3.37	3.85	3.00	3.33	3.66
Standard deviation	*1.71*	*0.51*	*0.89*	*1.30*	*1.30*	*0.57*
Understand software	3.00	3.12	3.71	2.50	3.08	3.00
Standard deviation	*1.73*	*0.35*	*0.95*	*1.06*	*1.24*	*1.00*

Table 5.27 Summary of questionnaire measures

Participants were also asked to state their view of the purpose of the modelling exercise. Responses were neatly split into two groups. One set correctly stated that the aim was to analyse data with a view to planning future databases and information provision (or similar). The second set less correctly identified the aim as reviewing and analysing processes (which is necessary to some extent in conceptual modelling but is not the ultimate aim). Only two respondents gave other reasons (meeting demand from members, advising members, gathering information for the IT Department).

					Understand	
Perceived purpose	**%**	**Valuable**	**Complete**	**Correct**	**Technique**	**Software**
Data analysis and planning	40%	4.16	3.50	3.50	3.66	3.00
Describe/review processes	40%	3.66	2.66	3.00	3.66	2.83
Form missing	7%	4.00	4.00	5.00	5.00	5.00
Meet demand from/advise members	7%	3.00	3.00	3.00	3.00	3.00
Obtain info. for IT Dept	7%	2.00	3.00	4.00	3.00	3.00

Table 5.28 Average questionnaire ratings by perceived purpose of modelling

Comments by participants

The questionnaires were not anonymous and, as in most business organisations, it is unlikely that strongly dissenting views would have been voiced because of this. However, most of the participants took the time to write comments on their questionnaires in support of their ratings, which suggests a degree of truthfulness. All of their comments are reproduced here. Text in brackets has been added for clarity. Organisation names have been replaced by ellipses.

There was generally a clear view of the purpose of the sessions and of the desired end result:

> *"(The) outcome will be a comprehensive model of information"*
>
> *"(The purpose is to) build a data model of what ... does/is responsible for"*
>
> *"(The purpose is to) define ... relevant information which may be of benefit to ... operators"*
>
> *"(The aim is) to establish/identify the different components and variables that would be required in developing a database"*
>
> *"To provide an infrastructure for handling of ... data"*
>
> *"To gather ideas on the kind of databases that should be controlled by the ... Association"*
>
> *"(To) help ... the Association in controlling and monitoring valuable information for its members."*

Four participants withheld judgement or were unsure what use the sessions would be:

> *"Interesting, useful but it will depend on the overall results achieved"*
>
> *"(The value of the information modelling sessions) will depend on the application of the data and the results achieved".*
>
> *"I am ... not sure what benefit (we) will derive from this exercise".*
>
> *"Overall I'm not sure that I found them (the sessions) useful. I would need to see the results of the following phases of the work."*

Several participants focused on the value of the modelling sessions in clarifying business processes as opposed to data:

> *"(The) focus is on what happens in your area ... the different tasks involved and who performs them. (It) clarifies (your) job ... and how that interacts with the rest of the organisation"*
>
> *"It explains our processes"*
>
> *"A unique experience in the business process"*
>
> *"(The sessions) identify component processes of translating member action into a valuable and accessible data base"*
>
> *"Having standardised processes can only assist the flow of information"*

But one participant felt that processes were not adequately modelled:

> *"It did not ... explain the work that is done, or where duplication takes place"*

Two participants highlighted common conceptual modelling problems:

> *"(It is) difficult not to go off on a tangent and discuss outside (the) necessary area"*
>
> *"Attempting to develop something definite out of something that wasn't very definite"*

Others would have liked the sessions to focus more strongly on the computer systems they currently use:

> *"It does not explain the association between the packages we use e.g. Excel and Opera"*
>
> *"As we were trying to anticipate all eventualities for the database it was difficult to relate them directly with the current function of the InfoCentre* (in-house information system) *... "*

One participant saw the sessions as irrelevant (*"I didn't learn anything new"*) and primarily useful for technicians (*"The IT Dept ... will gain a lot from it"*). Most other participants disagreed:

> *"(I) got a better outlook on documents and meetings"*
>
> *"(The modelling sessions were) well conducted, quite simple ... complemented by the tool ... (the) process is demystified and not hyped into something elaborate and over-complicated"*
>
> *"(Sessions were) very good, easy to follow ... useful is seeing how various procedures interact with each other and the organisation as a whole"*
>
> *"The session was well run and I found the general concepts quite interesting"*
>
> *"(The model) enabled us to view what we achieve in a simplistic way without being bogged down in detail"*
>
> *"(The sessions) help clarify (the) thinking process"*
>
> *"(The sessions were) hard! But extremely useful"*
>
> *"Useful exercise"*
>
> *"(The modelling sessions have been) useful to highlight areas which were not previously defined clearly"*

5.4 Observations by experimenter

Throughout the study the experimenter had many opportunities to observe both experienced and inexperienced modellers in action.

Getting started

Inexperienced method 'Y' modellers had more trouble getting started than inexperienced method 'X' modellers, perhaps because of 'blank page syndrome' (a blank page gives no clue as to how to start). All of the less experienced modellers required assistance to produce models. Less experienced method 'X' modellers were able to use this assistance and move on to finish their models autonomously. Less experienced method 'Y' modellers, in general, were not. It was difficult not to give greater assistance to the novice method 'Y' modellers since, without a certain level of help, they would have been unlikely to produce anything meaningful at all. These modellers often produced results that were incapable of interpretation (e.g. by drastically misusing the notation). In practice this meant that the experimenter had to point out significant problem areas in the models at an early stage of development, to ensure that the modellers were able to go on to generate measurable results. In the middle and later stages of modelling, questions from method 'X' modellers were answered, but most method 'Y' modellers were unaware that their models were significantly incorrect and, typically, did not ask questions. In fact, novice method 'Y' exhibited a remarkable, if misplaced, degree of confidence in their own work. Consequently, the experimenter was more likely with these modellers to take the initiative by pointing out areas that might need further work (but not demonstrating how to fix the models). Despite this, most method 'Y' models produced by non-experts failed to reach a nearly correct state.

The expert modeller (J) tended to get models right first time (using both techniques) without assistance. All of the less experienced modellers made relatively unstructured models at first, and then attempted to fix them. For method 'X', this process was probably made easier because of the tool support. For method 'Y', redrafting models was more onerous and the prospect of having to redraft tended to discourage the modellers.

The predefined set of types in method 'X' helped modellers by prompting them to look for concepts of each type. Method 'Y' modellers had no predefined concept types and were more 'at sea'. Most produced confused or notationally incorrect models. They tended to include ill-defined, generalised concepts that proved hard to refine, such as 'quality'. They also missed important aspects altogether (e.g. identifying people and organisations but failing to identify any activities).

Method 'X' modellers seemed to enjoy the experience of constructing models with words and pictures, and were apparently able to work this way intuitively. But, for method 'Y', the inexperienced modellers often appeared to lose concentration and forgot the meaning of their own model constructs, especially for more abstract concepts. The mental effort and discipline required to make sure the model meant the right thing were apparently too great. It seemed also that inexperienced method 'Y' modellers lacked the motivation to complete their models correctly – perhaps because they had no incentive to do so.

Graphics and annotations

Modeller J made extensive use of graphics in method 'X' to represent model concepts, and used a variety of background images. He hit upon the idea of colour-coding windows for quicker visual recognition. He also liked the ability to embed documents, and used 'concept' components to include annotations (e.g. documenting the terms of reference for modelling sessions). Strictly speaking, this practice was an error, since 'concept' components were intended to be used for structured business concepts, not for annotations. But it was an intentional misuse of the notation that greatly enhanced the value of the embedded documents, since related items could be included.

Therefore it was considered an improvement to the modelling technique and has not been treated as an error.

Modeller J created method 'Y' models that were conservative and traditional in style. But, with method 'X', he used the tool to create striking 'collage-style' models by combining pictures, embedded notes and other documents. For this and other reasons, modelling with method 'X' took on characteristics of its own that made it quite different from modelling with method 'Y'. Several of the less experienced method 'X' modellers also made use of graphics and annotations, including text documents, presentations and images, all of which seemed to provide useful context. Others did not bother much with either graphics or annotations—apparently seeing them as a waste of time (perhaps because they already knew what each concept represented, and were not expecting to have to communicate their models to others).

Since modeller J put in quite a lot of effort on both graphics and annotations, this might account for his somewhat reduced productivity relative to the other modellers when using method 'X'. It is not clear whether his time investment translated into quality improvements in the resulting models, but he felt that it made the process a more enjoyable and satisfying one for the participants, and he clearly enjoyed it more himself.

Activities

There seemed to be resistance amongst less experienced modellers (using both methods) against the idea of representing activity as a 'thing' (e.g. *purchase* as opposed to *purchasing*). Activity was seen as different from other concepts and inexperienced modellers wanted to represent it differently. Several modellers assumed activity was implicit and did not need representing at all. Inexperienced modellers often used types loosely, especially failing to recognise activities, which were often implicitly represented in other ways (e.g. the concept 'Lighting state', a change in lighting during a theatrical production, was represented in the 'Productions' model as a physical object).

Interpreting models

All modellers used the 'tree' view only infrequently, tending to alternate between the normal (icon) view and the interpretation view, with occasional forays into generated applications.

All method 'X' modellers (expert and inexperienced) made extensive use of the English language interpretation, which was clearly an essential tool for them. But several method 'X' modellers found it easier to judge if their models were correct by experimenting with applications produced by the tool's application generator.

Method 'Y' modellers could use neither interpretation nor generated applications, and less experienced modellers clearly were not 'on top of' their models, finding it difficult to interpret them and often surprised when the literal meaning was pointed out. They showed difficulty in grasping the concept or meaning of classes and associations, or even in some cases the idea that the model had a formal meaning at all. Informal or impressionistic interpretation was generally preferred, and working in the formal way demanded by method 'Y' did not come naturally. Less experienced method 'Y' modellers repeatedly tried to interpret several classes and relationships at the same time, attempting to understand the model 'as a whole'.

All experienced modellers, using both methods, showed signs of treating their categories as fuzzy, interpreting them in different ways at different times (although neither actually method supports alternative concept formation methods or fuzzy categories). Method 'X' allows the modeller to name components with different role names in different contexts, but the inexperienced modellers tended not to use this capability, possibly because it requires an analytical mode of thinking and some advance consideration of the issues before concepts are made concrete.

Notation and layout

In method 'Y' most inexperienced modellers failed to appreciate the distinction between attributes and classes, and often modelled concepts as both at the same time. No inexperienced method 'Y' modeller grasped the connection between relationships and attributes (foreign keys) and all produced contradictory models for this reason. In method 'X' this error also occurred but was less likely, probably because it was more easily identified. It is more obviously wrong to have two icons representing the same thing in the same window (e.g. both 'Customer' and 'Customer name' in a 'Purchase' window).

All inexperienced method 'Y' modellers misused the modelling notation, preferring their own more informal styles. Method 'Y' modellers paid attention to layout since diagrams quickly became hard to follow if not redrafted. This meant that the position of each class tended to change from one version to the next, making the model slightly less familiar. For several of the modellers, this led to a succession of model versions that were really quite distinct from one another, in both appearance and content. Often, correct ideas in one version were lost in the new version.

Method 'X' modellers tended to place components in a specific position and then leave them where they were. Hence the modeller typically could remember where to look for any given item. But method 'X' modellers had trouble finding components once windows had been closed. They often had to 'retrace their steps' (i.e. follow a series of relationships), opening a series of windows to find a component.

Categories

Method 'X' modellers seemed intuitively to understand model components where the default picture was 'obvious' (person, organisation, place, document, etc.). But most modellers who used method 'X' did not distinguish effectively between the two types 'category' and 'conceptual object' and used them interchangeably. This is perhaps not too surprising since these are the least concrete of the nine available business concept types.

In general, less experienced modellers did not seem to distinguish things in a clear-cut way from their categories. For example, in the Purchase Orders model, the modeller used concept 'Product' to represent both the product code (i.e. product type, a category) and the product itself (i.e. product, a physical object).

Only the expert modeller (J) introduced abstract concepts (mainly as supertypes for ease of modelling).

5.5 Summary

This chapter has presented the findings from the experimental part of this research, described in Chapter 4. The experiment was divided into a main study, comprising nineteen separate modelling exercises with comprehensive analysis, and a secondary study with a larger number of modellers but less detailed analysis. The results of the main study presented here include a detailed numerical analysis of models, with model development tracked in detail for each model. Participant observations and questionnaire responses provided by modeller and group session participants are also detailed. Interview notes are reproduced in full in Appendix C and an interpreted version of the interview notes is presented in Chapter 6. The results of the secondary study comprise statistics on model quality. Together, the two studies provide a rich data set offering many opportunities for exploratory interpretation, cross-comparison and triangulation. Chapter 6 offers a full interpretation of the experimental results.

6
Interpretation of Results

Chapter 5 presented several collections of data arising from the experimental part of this research. The data includes questionnaire responses, participant observations, and the results of numerical model analysis. In addition, interview notes are presented in Appendix C. In this chapter we piece together the various experimental results from Chapter 5 and Appendix C to form a composite picture of what happened when modellers of different types used each method. The aim is to try to understand and interpret the results from the modellers' own points of view. The results of the experiment and the interpretations placed on those results are used to generate a number of more general theories about conceptual modelling practice.

6.1 Analysis of experimental results

The following summary of observations by the experienced modeller is based on the interview notes, which are reproduced in full in Appendix C.

Observations by experienced modeller

Despite having prior reservations about not being able to use method 'Y', with which he was very familiar, modeller J ultimately expressed a preference for method 'X' over the traditional approach. He reported that method 'X' was more mentally taxing (*"in respect of knowing where you are"*) but stated that he preferred it because it seemed to be easier for the participants to understand and use. This was explained as a result of *"the simplified view—the graphic view is more accessible than a diagrammatic view that the participants are probably not familiar with"*. According to modeller J the participants were *"intuitively ... more in touch with the picture—they are not faced with a big wiring diagram, they are digesting it in chunks"*.

With method 'X' modeller J found he had to call breaks periodically, to review the model. The participants went for coffee while the modeller checked the model in detail. *"If you have a method 'Y' model on the board you can quickly see where the weak areas are, and where the relationships are. But with method 'X' you tend to follow a line from component to component, and not really know if you are missing another line somewhere else"*. He indicated that this carried certain risks but was not necessarily a bad thing (*"there is nothing wrong with calling regular breaks"*). However, he also predicted that it would be difficult for an inexperienced modeller to find areas in a model that were weak and needed more work, especially if the business were unfamiliar.

Modeller J reported that more was achieved with method 'X' in the modelling sessions than when using method 'Y', although he offered no firm evidence for this. But he also felt that the method 'X' models probably needed more work after each session.

Overall, modeller J reported that the quality of the models produced with method 'X' was better, which he attributed to improved understanding by the users. According to modeller J, the act of classifying each concept (as a person, organisation, document, …) aided understanding, got the participants engaged in discussion and helped them clarify their ideas about what each concept actually meant (or could mean). *"It almost doesn't matter what categories you have to choose from—it just helps to have categories so you have to have a discussion".* Consequently descriptions in method 'X' models were more accurate and the attributes better thought out. *"With method 'X' you are asking a more specific question—"what type is ...?"—so you can get a more specific answer, not just general agreement to a description".*

Modeller J perceived several other aspects of method 'X' as useful. Being reminded of existing components was not particularly valuable for the modeller, but it was useful for the participants. Choice of colour and backgrounds assisted in recall, and the ability to use impressive audio-visual material helped to increase arousal (it provided, in his words, *"a visual jab in the ribs"*). Modeller J found the English-language interpretation an essential tool, despite his expert knowledge of modelling, since it reduced the mental effort required to verbalise the model and made its meaning concrete. Using the interpretation also provided some structure to the modelling sessions, by increasing the need for periodical breaks.

Questionnaire responses

It is clear from the responses that the participants generally understood the nature of the process they were part of, both for method 'Y' and for method 'X'. For method 'Y' two specific modelling problems were mentioned: *"(It is) difficult not to go off on a tangent and discuss outside (the) necessary area"* and *"(the exercise involved) attempting to develop something definite out of something that wasn't very definite".* No method 'X' participant voiced reservations of this nature about the modelling technique. However, it would probably be unsafe to read much into this fact. The analysis of questionnaire responses was limited by the low number of questionnaires available, and overall there seemed to be few qualitative differences between the reported experiences of group members who used either method.

Analysing the ratings by the individual characteristics of the group members throws up some interesting results. Those who had created a database before were slightly more likely to judge the resulting model as complete and correct. The value of the sessions was also rated more highly by those who had created databases before and those who were educated to diploma or degree level. Those who participated in a second (i.e. follow-up) modelling session were slightly more likely to judge the model as complete and correct (as one would hope).

Respondents who correctly identified the purpose of the modelling session were also more likely to judge the models as complete and correct and to rate the sessions as valuable. 57% of those who had experience of creating a database were able to correctly state the purpose of the sessions. Only 25% of those who had no experience of creating a database could identify correctly the purpose of the modelling sessions. This suggests that having created a database at some point in the past gives one insight into the reasons for modelling. Respondents who had never created a database before were twice as likely to state that the purpose of the sessions was to model processes rather than data. This apparent slight misapprehension on the part of approximately 45% of participants did not seem to affect the outcome of the modelling sessions, however.

Observed completeness and correctness

Measures for completeness and correctness were calculated by comparing each finished model with a further, corrected version. In the graphs that follow each model has been allocated to a group according to the modeller's experience level and modelling technique, as shown in Table 6.1. Measurements for less experienced method 'X' modellers are circled in red, with those for method 'Y' modellers in green.

Technique	Modeller's experience level		Group
Method 'Y'	Inexperienced/novice	No experience	1
		Little experience	2
	Experienced/expert	Very experienced	3
Method 'X'	Inexperienced/novice	No experience	4
		Little experience	5
	Experienced/expert	Very experienced	6

Table 6.1 Modeller groups

It was observed that method 'X' models were on average more complete than method 'Y' models—all method 'X' models were more than 80% complete. No method 'Y' model produced by a non-expert was more than 40% complete, apart from one produced by a modeller with some prior experience. For inexperienced modellers the final completeness of the models was apparently related to the level of prior modelling experience. There was also far greater variability in the completeness of method 'Y' models. The expert modeller produced 100% complete models (or as nearly complete as could be measured) using both techniques (excluding model 4, which was abandoned at an early stage of development for operational reasons, and has therefore been omitted from the analysis in this chapter). Correctness showed a similar pattern. The final correctness of the method 'X' models was above 80% for all but one model, whereas method 'Y' models produced by inexperienced modellers were all 40% correct or worse. Most were less than 25% correct. For comparison, in the secondary study the average completeness achieved by modellers using method 'Y' after some training was 43% (shown for convenience in yellow in Figure 6.1), which is in line with the results of the primary study.

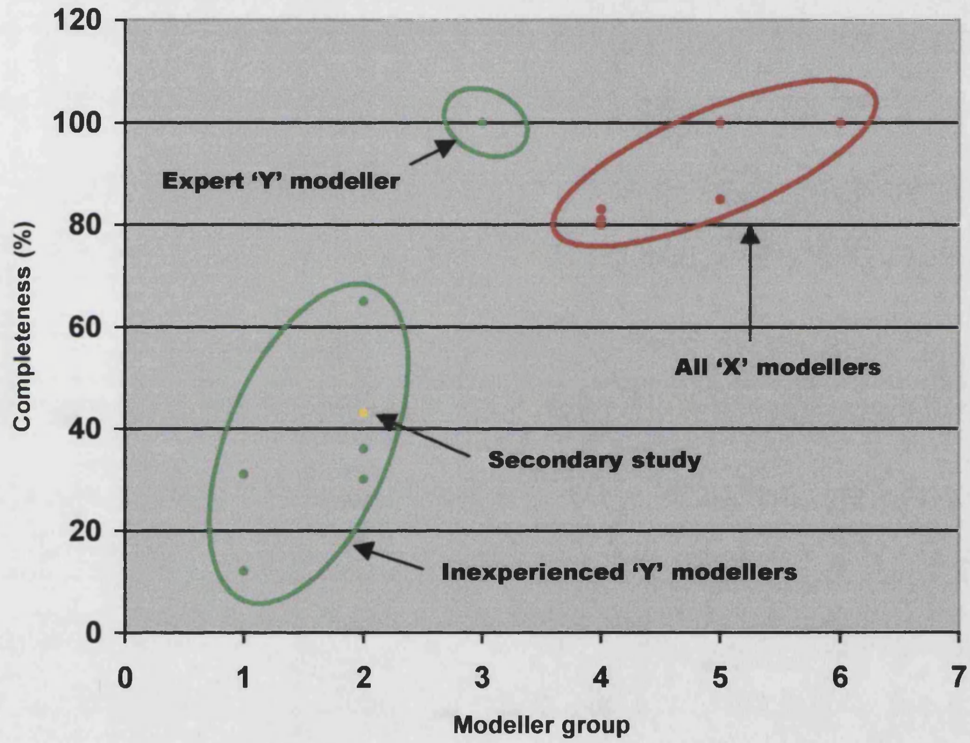


Figure 6.1 Completeness vs. modeller's experience level

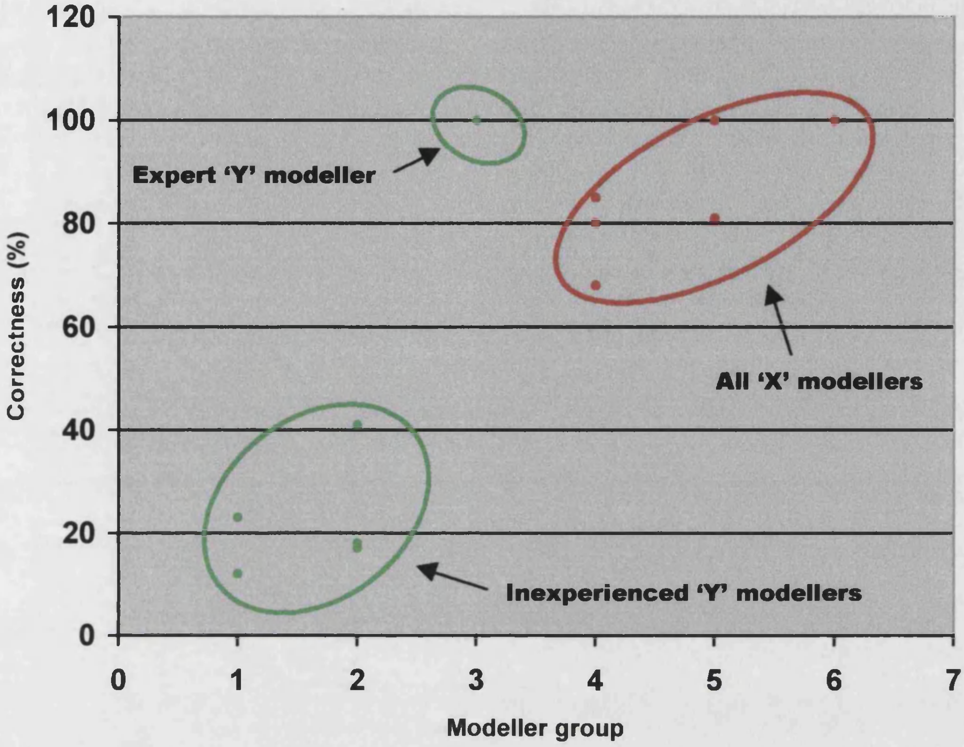


Figure 6.2 Correctness vs. modeller's experience level

Observed productivity

Both experienced and inexperienced modellers were more productive with method 'X' than method 'Y', with an average difference in productivity of 147% for the expert modeller and 366% for inexperienced modellers. All but one of the method 'X' models were produced at a rate of over 3.5 components per hour. In contrast, most method 'Y' models were produced at a rate of less than 1.5 components per hour. This means that a method 'Y' modeller will typically take between twice and five times as long to produce the same model. Using method 'Y', the productivity of inexperienced modellers fell significantly short of that of the experienced modeller. But for method 'X' productivity was more uniformly high for experienced and inexperienced modellers.

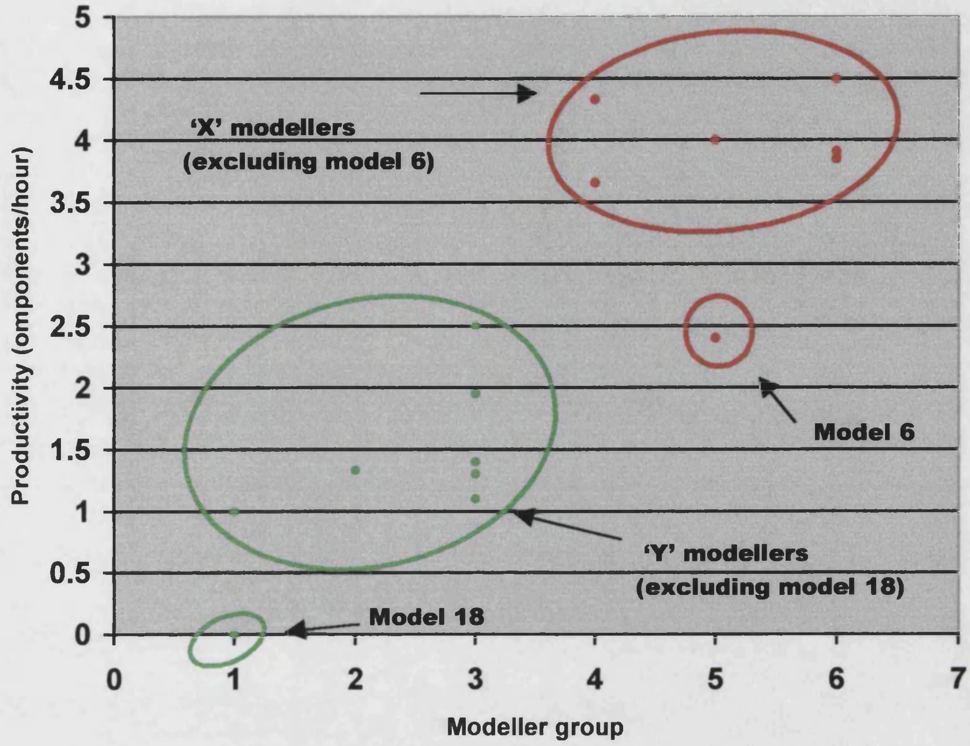


Figure 6.3 Productivity vs modeller experience level

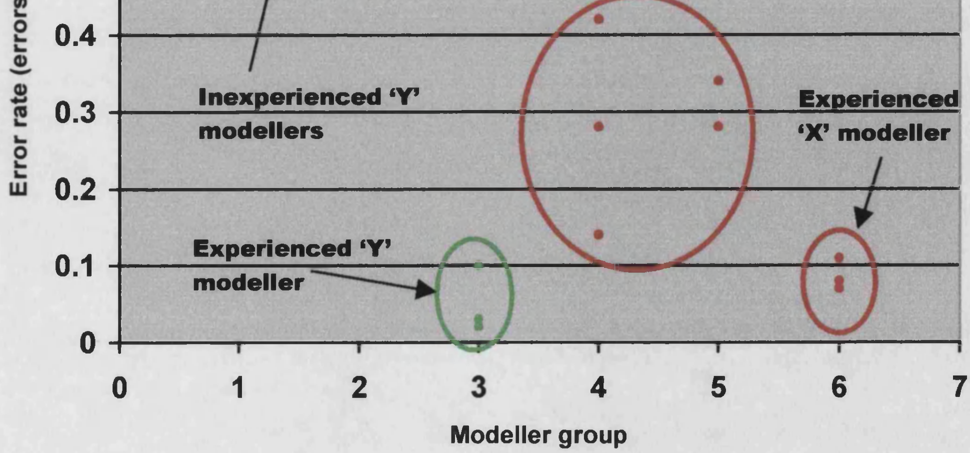


Figure 6.4 Error rate vs modeller experience level

Errors

The experienced modeller's error rate was uniformly low for both modelling techniques. However, for less experienced modellers the average error rate using method 'Y' (0.59 errors per change) was high in comparison with that achieved with method 'X' (0.29). All but two method 'X' models achieved an error rate below 0.3. In contrast, no method 'Y' model achieved an error rate below 0.5 except those produced by the experienced modeller. This means that on average nearly two-thirds of the actions performed by inexperienced modellers using method 'Y' were mistakes. In contrast, less than one-third of actions performed by inexperienced modellers with method 'X' were mistakes.

Table 6.2 summarises the relative error frequencies for the two modelling techniques. As for changes (Table 5.24) there is remarkably little overall difference between the distributions of error types for the two techniques (correlation .997). By far the most common type of error is the creation of incorrect relationships (an incorrect relationship is one which is either completely redundant or where either or both participating components are wrongly chosen). A less frequent, but still significant, source of error is the creation of incorrect components. This includes components named wrongly and redundant components. Other sources of error (e.g. incorrect relationship cardinality) occur far less frequently.

Error	**Method 'Y'**	**Method 'X'**
Incorrect relationship added	72.79%	69.24%
Incorrect component added	20.44%	24.55%
Relationship wrongly removed	3.30%	2.40%
Cardinality altered wrongly	2.04%	1.64%
Component wrongly removed	1.41%	2.15%

Table 6.2 Error frequencies

Less consistency is apparent when we look in more detail at the breakdown of errors according to experience. The graph below shows the relative frequencies of different errors made by the modellers in each group.

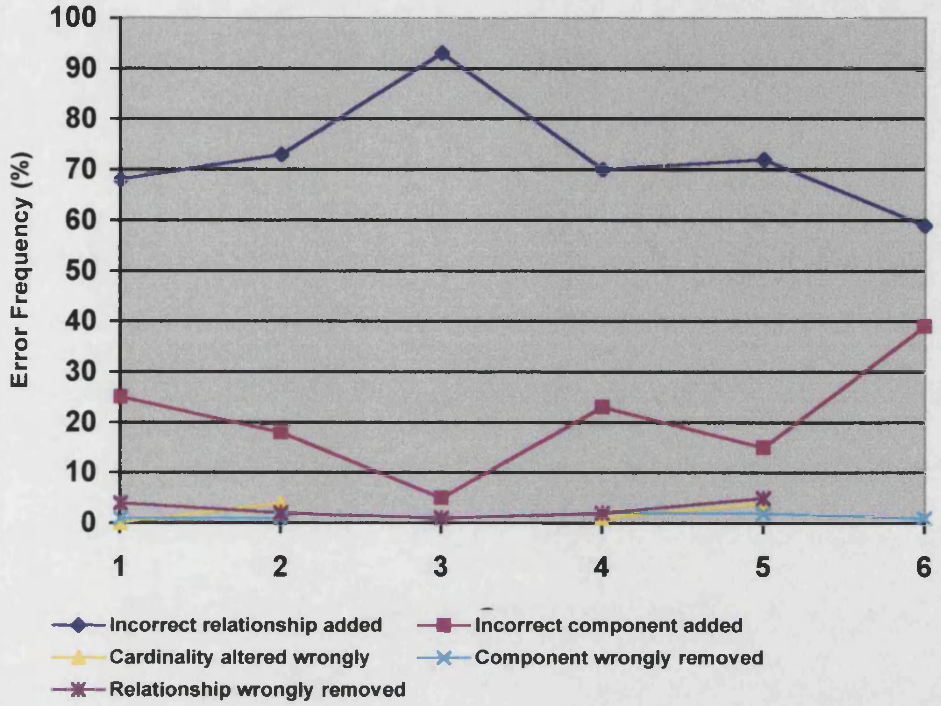


Figure 6.5 Error frequencies

The feature most of interest in the graph is the disparity between the mix of errors for the expert modeller when using method 'Y' from that when using method 'X'. The error frequencies are roughly constant for the 'no experience' and 'some experience' groups (i.e. groups 1, 2, 4, and 5) across the two modelling techniques. But in the case of the expert modeller (groups 3 and 6) the ratios are quite different. When using method 'Y' the expert modeller produced many incorrect relationships (93%) and few incorrect (5%) components. But with method 'X' these two types of error are more evenly balanced (60% incorrect relationships and 40% incorrect components).

An analysis of the expert modeller's errors sheds some light on this. The number of incorrect relationships is constant between the two techniques, while the number of incorrect components is a good deal higher for method 'X'.

Error	Method 'Y'	Method 'X'
Component wrongly removed	0	1
Incorrect component added	3	37
Incorrect relationship added	56	56
Relationship wrongly removed	1	0

Table 6.3 Error frequencies for expert modeller

Calculating effectiveness

The effectiveness level is a rough indication of relative overall modelling success and takes into account the modeller's productivity and average error rate, and the model's overall correctness and completeness. Productivity figures were not available for some models since the total modelling time was not recorded (see Table 5.21). Hence no 'effectiveness' figure has been calculated for these models.

Method	Modeller's experience	Effectiveness	Models
Method 'X'	None	Poor Good	19 7
	Some	Poor Good	6 1
	High	Very good	9, 10, 16
Method 'Y'	None	Very poor	8, 18
	Some	Very poor	3, 14
	High	Poor Good	11, 13, 15, 17 12

Table 6.4 Summary of effectiveness levels achieved

Overall, there was strong correlation (average .95) between the model quality measures (completeness, correctness and error rate) used to calculate effectiveness. Hence these measures can be regarded as facets of the factor 'effectiveness'. Productivity was correlated strongly with these measures (.92) for the less experienced modellers, reflecting the fact that model quality was universally high for the expert modeller irrespective of other factors.

Complexity was inversely correlated with all other measures. This effect was only slight for the expert modeller, but marked for the inexpert modellers (.73). It is obviously harder to model more complex areas correctly, but this result demonstrates that complexity is especially a problem for less experienced modellers.

	Complexity	Completeness (%)	(1-Error rate)	Productivity
Correctness	-0.44	0.98	0.95	0.54
Complexity		-0.38	-0.34	-0.48
Completeness			0.92	0.52
(1-Error rate)				0.39

Table 6.5 Correlation of effectiveness facets for all modellers

	Complexity	Completeness (%)	(1-Error rate)	Productivity
Correctness	-0.81	0.96	0.94	0.93
Complexity		-0.64	-0.78	-0.87
Completeness			0.88	0.85
(1-Error rate)				0.98

Table 6.6 Correlation of effectiveness facets for inexpert modellers

According to the effectiveness calculation given in Chapter 5, the average overall effectiveness of modellers using method 'Y' was *poor* (for the most experienced modeller) or *very poor* (for the inexperienced modellers), whilst that of modellers using method 'X' ranged from *good* (inexperienced modellers) to *very good* (most experienced modeller). The expert modeller was less effective when using method 'Y' (four 'poors' and one 'good') than when using method 'X' (all 'very good'). This is not an indication that the method 'Y' models were faulty but reflects the fact that they took a good deal longer to produce. Figure 6.6 plots the relative effectiveness of the modellers in each group.

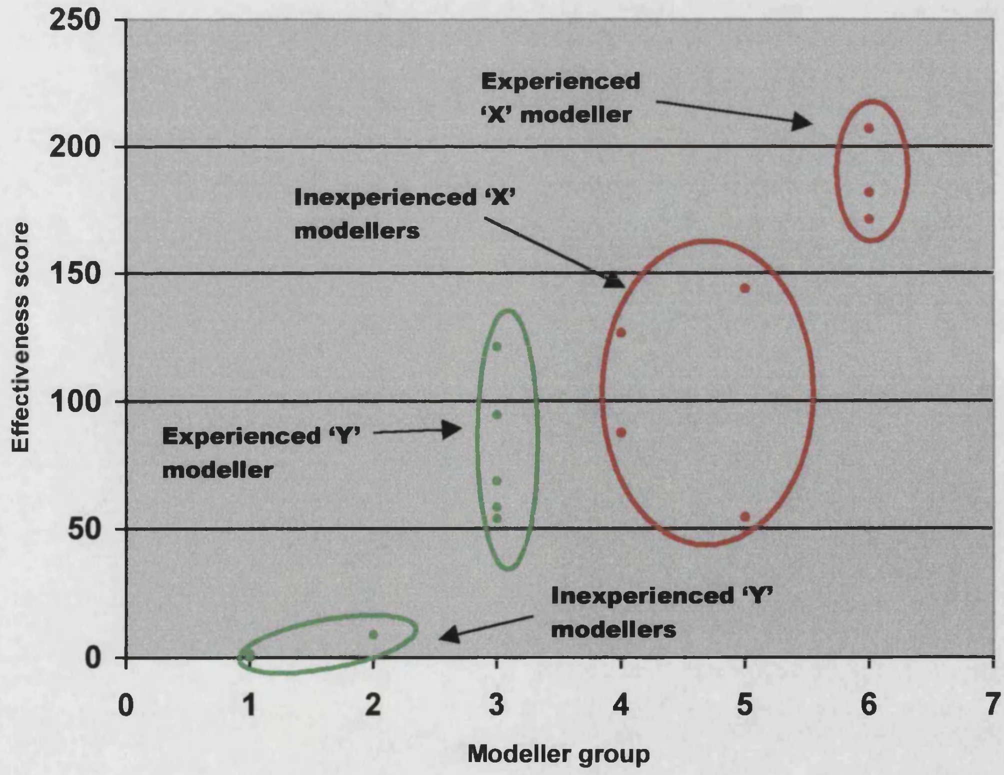


Figure 6.6 Effectiveness vs modeller experience level

Are the effectiveness figures meaningful? There are probably alternative ways of calculating effectiveness that would give different results. However, any method that took into account the four factors (completeness, correctness, error rate and productivity) would be subject to the same underlying trends. The best and worst effectiveness scores (expert modeller with method 'X' and inexperienced modellers with method 'Y' respectively) come as no surprise because they tally with the observed performance of these modellers during the experiment. The fact that the expert modeller's effectiveness scores when using method 'Y' were lower than the inexperienced modellers' scores when using method 'X' is perhaps surprising but, given the arbitrary choice of effectiveness calculation and the small sample size, it would probably be prudent to avoid reading too much into this particular result.

Estimating usability

The difference between effectiveness levels for experienced and less experienced modellers is 68.45 for method 'X' and 76.03 for method 'Y' (average 72.24). Since these figures are similar we might reasonably take them as a numerical approximation for the average difference in abilities between the experienced and inexperienced modellers. The equation:

Usability of method = Effectiveness of modeller using method - Ability of modeller

was introduced in Chapter 5 as a way of calculating a very rough indication of relative usability. The theoretical maximum effectiveness is 200. Table 6.7 summarises the average effectiveness scores achieved as percentages of this maximum.

	Inexperienced modeller	**Experienced modeller**	**Difference (ability/experience)**
Method 'Y'	2% (±2)	40% (±14)	38% (±14)
Method 'X'	52% (±20)	86% (±25)	34% (±25)
Difference (usability)	50% (±20)	46% (±25)	

Table 6.7 Rounded average effectiveness scores (as % of theoretical max. effectiveness)

From this analysis, the contribution of experience (ability) appears to amount to about 34-38% of an average modeller's overall effectiveness. The contribution of method 'X' over method 'Y' appears to account for about 46-50% of overall effectiveness. Hence we could conclude that the introduction of psychological ideas in method 'X' seems to have a very significant impact on overall effectiveness, for both the expert and the less experienced modellers. The effect is significantly greater than the effect of training and prior experience (approximately 33% greater, according to this calculation).

6.2 Interpretation

The results of the experiment seem to point to a conclusion that, under the right circumstances, inexperienced modellers with little or no prior experience can perform as well as an expert modeller with many years' training and experience. Despite the limited sample size, quite compelling results were obtained. Using method 'X', people with little or no prior conceptual modelling experience produced models of near-expert quality, more quickly than an expert. The models were smaller than those produced by the expert but still of a realistic and useful size. It would be wrong to take the experimental results too literally. The experiment was not a controlled study in any scientific sense and it proved nothing in absolute terms. But the body of evidence, including reasoned argument (Chapter 3), quantitative data from analysis of models (Chapter 5) and observations by modellers and experimenter (Chapters 5 and 6), seems to support the idea of applying psychological principles to the conceptual modelling problem. The experiment did not allow us to seriously investigate which, if any, of the psychological principles established in Chapter 4 gave rise to the apparent improvements in performance. But it provided some clues. In the remainder of this chapter we shall consider the psychological principles in the light of the results of the study.

Conceptual modelling, as it is currently practised, can be viewed as an obfuscating barrier to end-users, a tool used by IT professionals to retain control over resources and processes. This thesis is, in some ways, about the empowerment of end-users who would like access to the advantages that being able to produce well-designed information systems can bring. The radical goal of this research is to find out if reliance on experienced IT professionals (and on expensive packaged software products) can be reduced. One way to achieve this goal is to see if the currently-accepted design process can be simplified, to the extent that end-users can perform it quickly, easily and effectively, with little or no need for assistance from trained IT specialists.

Hence this research is, in part, about helping the inexperienced emulate the expert. An expert has predefined frameworks for thinking about problems, and internalised skills that can be applied without the need for conscious thought. Neurolinguistic programming (NLP) introduced the idea of behavioural modelling, in which an expert's modes of behaviour and thought are mapped so that non-experts can apply them (Dilts et al 1979). We have approached the same problem using three main strategies: (a) adjusting the content of models so as to obtain a good match with mental concepts, (b) representing models in such a way that people can understand them more easily, and (c) making the modelling process simpler and easier (this set of three categories corresponds roughly with the division of concerns used in Chapter 3—content, representation and process). Below we consider each of these strategies in turn. However, we shall first examine the role of expertise in conceptual modelling, and assess the overall results of the experiment in this context.

The nature of expertise

Traditionally, conceptual modelling has been based on some implicit assumptions about expertise. One assumption is that modellers can and will become expert before attempting to create models: a key belief behind the design of most modelling techniques is that the modeller will be a well-trained specialist who embarks on modelling only once he or she is armed with the requisite skills. Hence method designers do not concern themselves overly with making their techniques simple or easy. Instead, they focus on issues such as formality and economy of representation, and rigour (e.g. Gregory 1995, Herbst 1997). And once the hypothetical expert modeller has started modelling, the assumption is that he or she will continue until the model is of good enough quality before attempting implementation.

The first assumption, that only experts will produce models, is at best optimistic and is probably unwarranted. Highly experienced data modellers do not produce most data models. The continued shortage of trained staff in the IT industry is only one reason why fully-trained are often not available or used. Many millions of copies of "personal database" products like Microsoft Access have been sold. Many untrained end users struggle with these products and a proportion succeed in producing "good enough" systems that they can use. Although they may not consciously set out to do conceptual modelling, that is inevitably what they have to do to produce useful database applications.

The second assumption, that modellers will continue to work on their models until they are correct, before implementing them in databases and applications, is also rather suspect. Experience suggests that common industry practice in this respect leaves quite a lot to be desired. Untrained or inexperienced staff who produce models may be unaware of most of the mistakes they make, and therefore unable to fix them. Time pressures also prevent sufficient time being spent on models.

The inexperienced modellers in this study produced models that were approximately 80% complete and correct when using method 'X'. Is 80% good enough? It may be a fallacy that only completely correct models can be implemented. The results of this study would certainly suggest that object modellers of 'average' ability produce models that are a good deal less than 100% complete and correct, although the modellers in this study cannot be taken as representative of the range of skill levels at large in the IT industry. We have been taught to think about information systems analysis, design, and use as distinct activities. In reality, this received wisdom was a response to the high cost of correcting design flaws. It is often considered essential to spot mistakes before the design stage, and well before implementation, so as to avoid having to make expensive modification to systems when problems come to light during later use. But the popularity of RAD and prototyping has demonstrated that sometimes it is better, easier, and even cheaper to spot errors by working with implemented systems, or prototype versions of them. If the cost of correcting errors in systems can be reduced sufficiently, then implementing models that are only 80% complete and correct may not be a disadvantage. In fact, it may allow users to assess models in a very effective and efficient way: by living with their consequences.

Patterns of expertise

It became clear during the course of this study that three distinct patterns occurred repeatedly in the evolution of models. The three patterns are described below.

Pattern I: Experienced modeller using either technique The typical pattern for the experienced modeller (using either technique) is given in Figure 6.6. Models 9-13 and 15-17 exhibit this pattern. As can be seen, the error rate declines rapidly and tends towards zero, and accuracy is high.

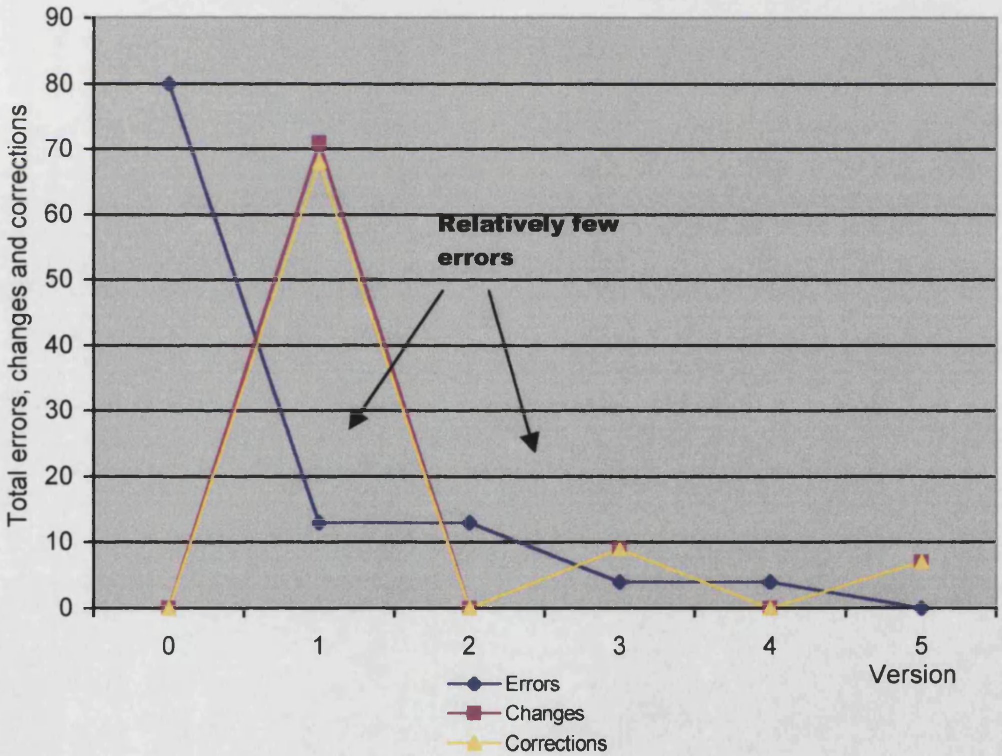


Figure 6.7 Typical pattern for an experienced modeller using both methods

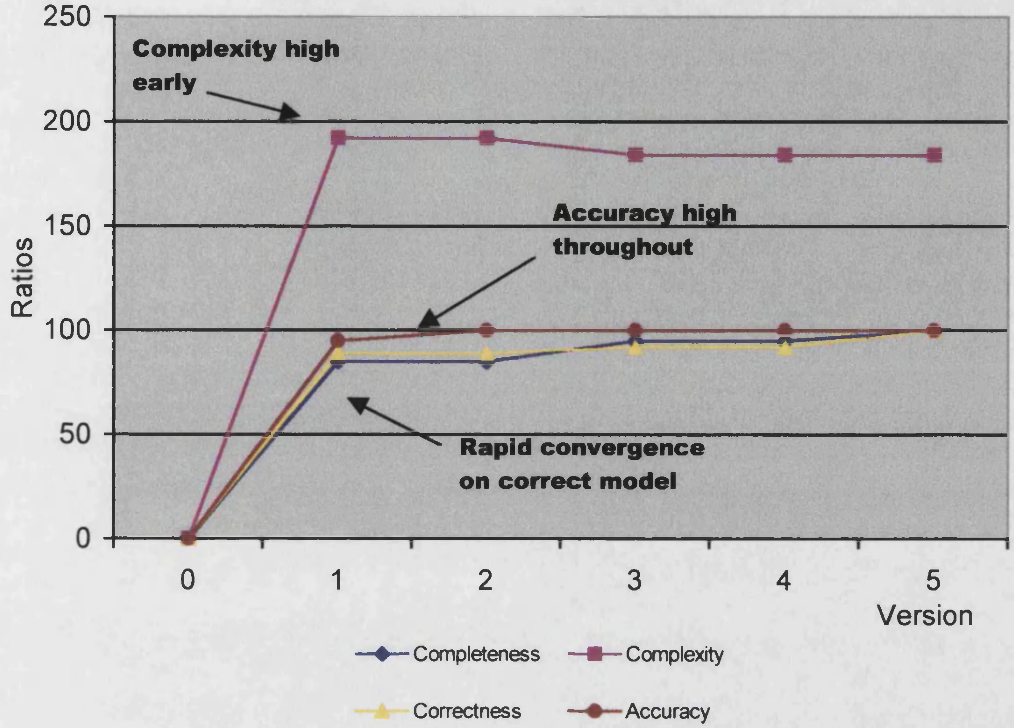


Figure 6.8 Typical pattern for an experienced modeller using both methods

Completeness and correctness quickly attain levels close to 100%. Complexity is high throughout and declines slightly as the model is completed.

Pattern II: Inexperienced modeller using method 'X' A roughly similar pattern was seen for inexperienced modellers using method 'X' (Figure 6.7). Once again, the error rate shows a decreasing trend over the course of development of the model. The overall completeness of the model increases smoothly. However, after the initial creation phase of the model the proportion of errors remains roughly stable ("plateau" phase), primarily because of unwanted or incorrect relationships. Eventually the modeller rectifies the errors and completes the model. The model does not become over-complex and quickly reaches nearly its final level of complexity. Completeness and correctness show overall increasing trends. The modeller's accuracy fluctuates but improves overall as the modelling progresses and ends high. Models 1, 5-7 and 19 exhibit this pattern.

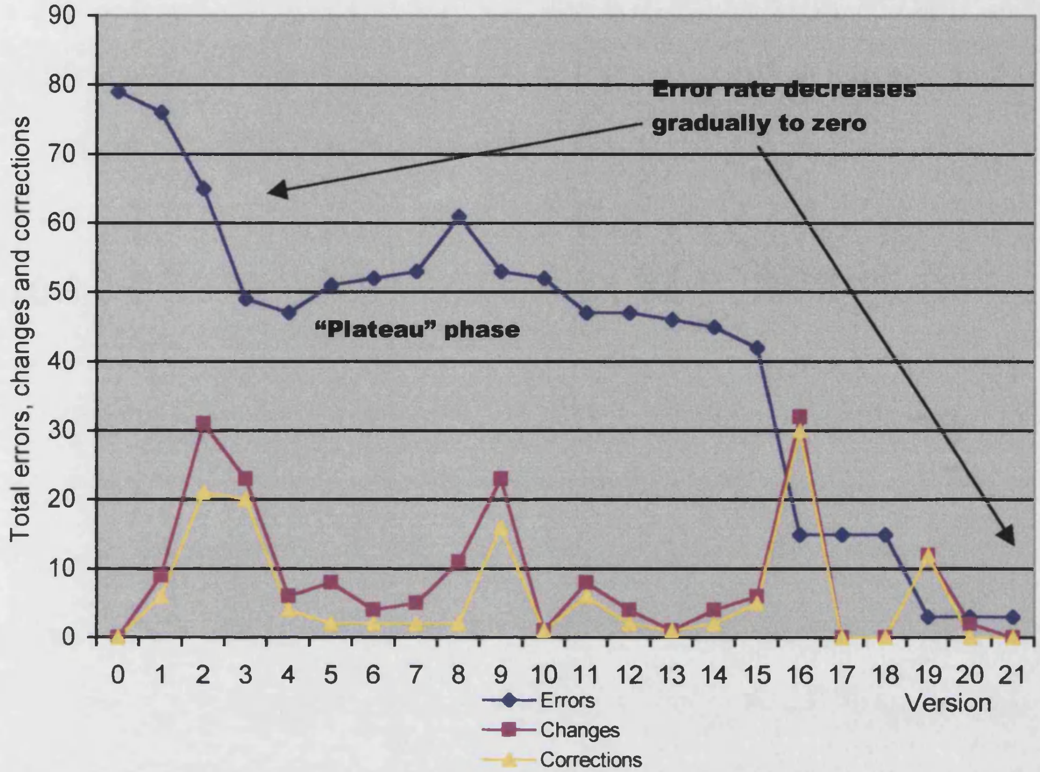


Figure 6.9 Typical pattern for an inexperienced modeller using method 'X'

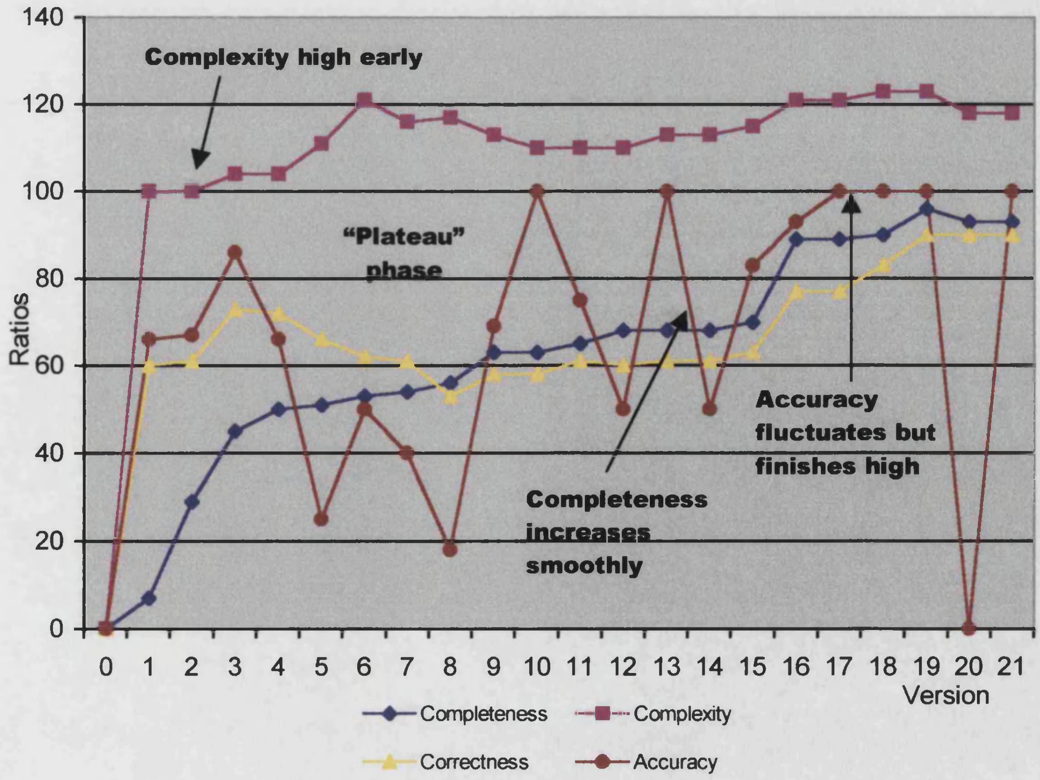


Figure 6.10 Typical pattern for an inexperienced modeller using method 'X'

Pattern III: Inexperienced modeller using method 'Y' For an inexperienced modeller using method 'Y' the typical pattern is quite different from either of the preceding two patterns, as seen in Figures 6.11 and 6.12. In these models the error count increases to a point where the modeller either gives up or wrongly judges the model to be complete. Complexity increases until it becomes significantly higher than it should be. Completeness and correctness fail to increase convincingly and may even decline (note that the corrected version of the model has been omitted from Figures 6.11 and 6.12 for clarity). Accuracy is poor. Models 2, 3, 8, 14 and 18 exhibit this pattern.

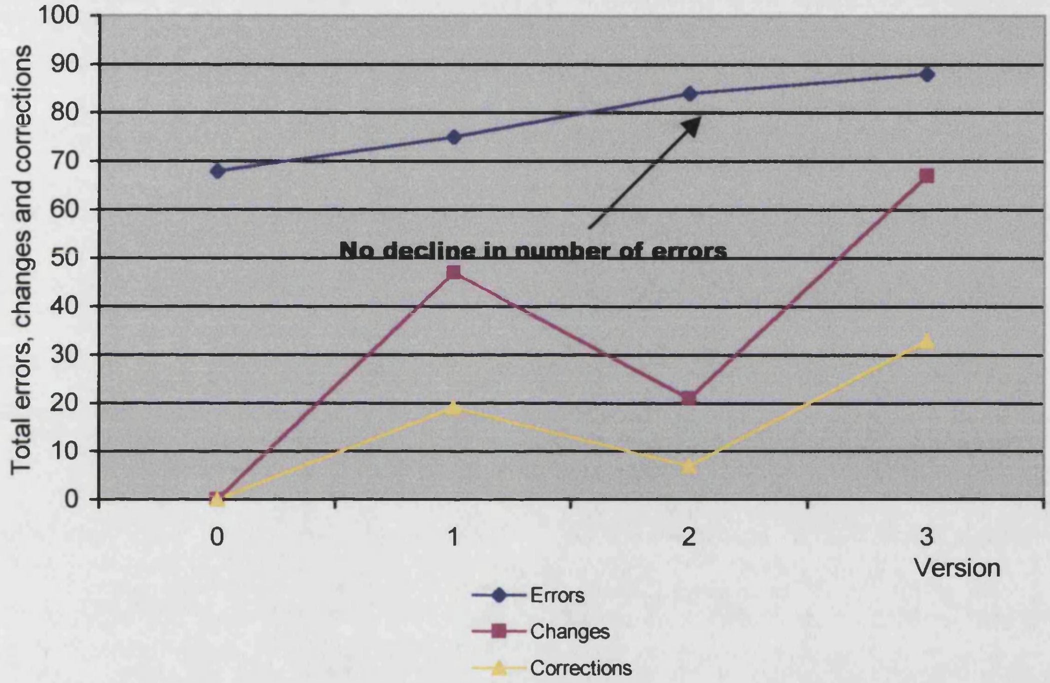


Figure 6.11 Typical pattern for an inexperienced modeller using method 'Y'

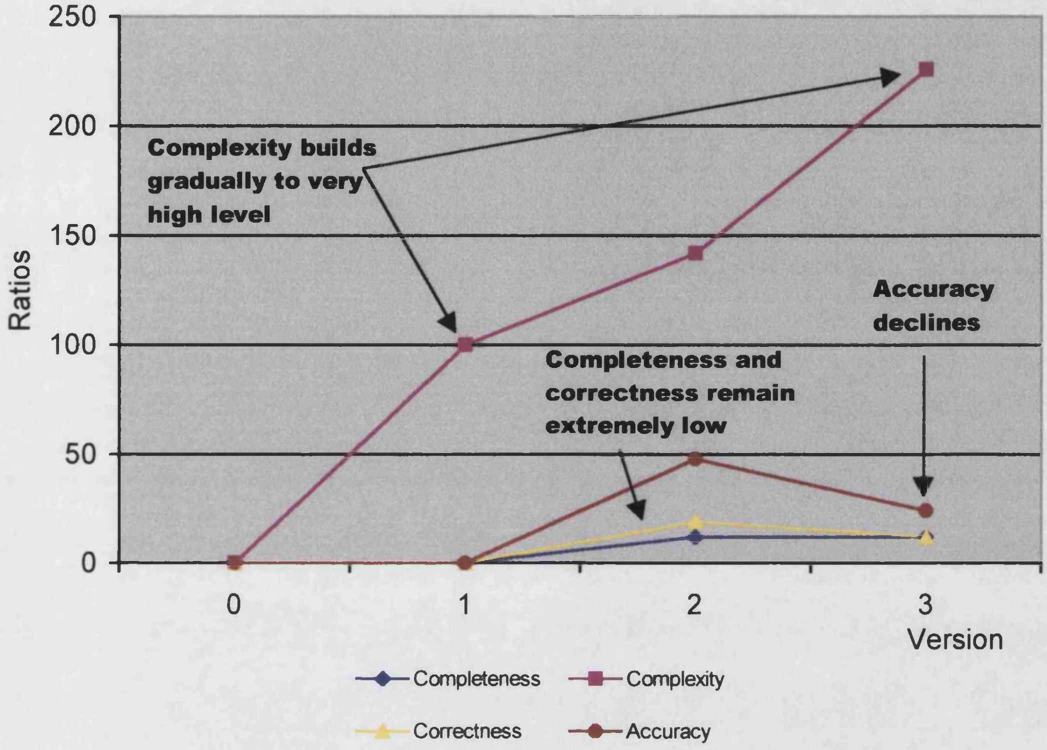


Figure 6.12 Typical pattern for an inexperienced modeller using method 'Y'

Model 18 is a good example of pattern III. The inexperienced modeller creates a model that does not make sense because it is not structured meaningfully. More or less correct classes are identified but few, if any, of the relationships are correct. Models of this nature were generally not completed because the modeller did not appreciate that the model was faulty.

The patterns observed for different skill levels in each technique are summarised in Table 6.9. Some similarity can be seen between pattern I (experienced modeller using either technique) and pattern II (inexperienced modeller using method 'X'). An inexperienced modeller cannot behave in the same way as an expert since he or she typically does not know in advance how best to model. An expert can go straight to the best solution, and as we might expect the typical curve for the expert (pattern I) shows a steadily decreasing error level. The less experienced modellers using method 'X' tended to increase total errors in the early stages of modelling (pattern II). But they were able to recognise and correct these errors so that, overall, they ended up with a decreasing trend in error level. Inexperienced method 'Y' modellers simply created more and more errors (pattern III), and were not able to correct their models. Based on the points of similarity between patterns I and II we can claim that method 'X' helps less experienced modellers behave like experts in several important ways.

	Experienced modeller	Inexperienced modeller	
	Methods 'X' and 'Y' (Pattern I)	Method 'X' (Pattern II)	Method 'Y' (Pattern III)
Error level	Decreases steadily	Possible initial or mid-term increase followed by decreasing trend	Steady increase without eventual correction
Complexity	Increases to final level	Rapidly increases towards final level and remains roughly constant	Increases drastically to well above required level
Accuracy	Consistently high	Fluctuates, tending towards 100%	Fluctuates, remaining low
Completeness	Consistently high	Tends towards 100%	Remains low
Correctness	Consistently high	Tends towards 100%	Remains low
Incorrect relationships	Few, eventually corrected	Many, eventually corrected	Many, not corrected
End result	Complete and correct	Largely complete and correct	Largely incomplete and incorrect

Table 6.9 Patterns observed for experienced and inexperienced modellers using both methods

Table 6.10 shows the typical levels of productivity and effectiveness for each type of modeller.

	Experienced modeller		Inexperienced modeller	
	Method 'X'	Method 'Y'	Method 'X'	Method 'Y'
Pattern	I	I	II	III
Productivity	Very high	Low	High	Very low
Effectiveness	Excellent	Poor	Good	Very poor

Table 6.10 Summary of typical performance for each type of modeller

Other evidence of patterns

For all modellers and both modelling techniques, the number of relationships in a model was approximately proportional to overall model size with a ratio of very approximately 2. Figure 6.13 illustrates this relationship.

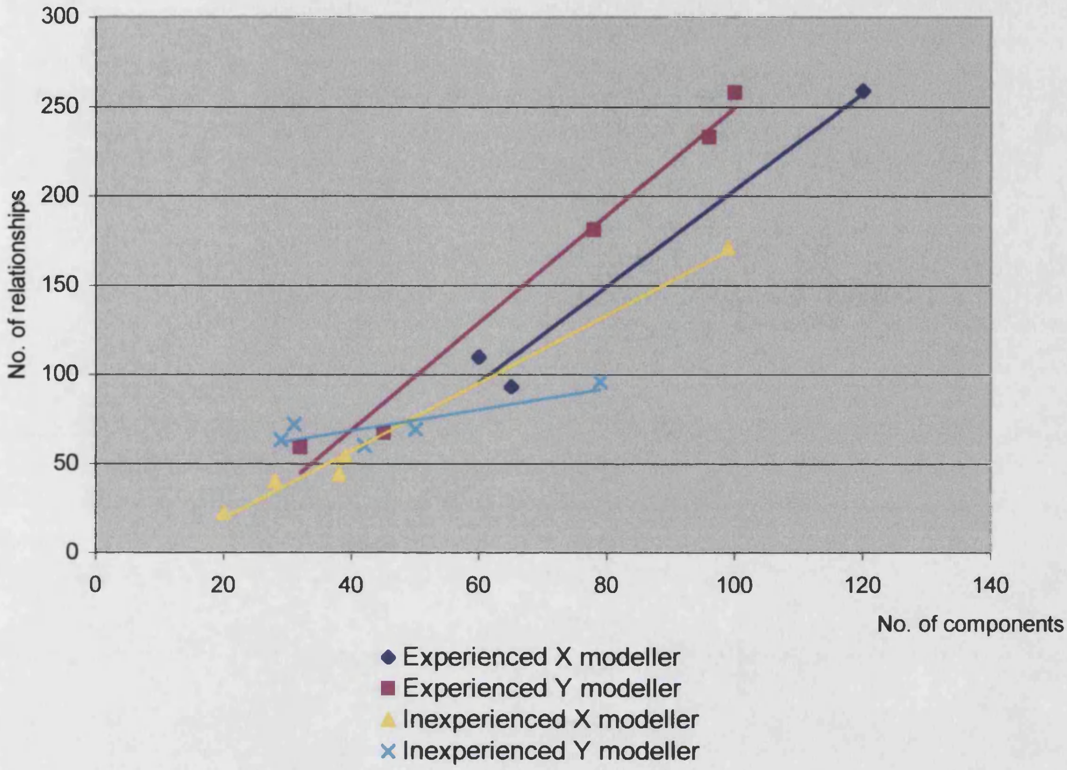


Figure 6.13 No. of relationships vs. no. of components

For inexperienced 'Y' modellers, the ratio of relationships to components was somewhat lower than for the expert modeller. For inexperienced 'X' modellers, the ratio was closer to that of the expert modeller, perhaps adding weight to the view that method 'X' allowed inexperienced modellers to perform in a way that was more like the expert.

The number of changes made by modellers was clearly related to both the complexity of the model (number of relationships) and to the modeller's level of expertise. Figure 6.14 shows four trend lines corresponding to experienced and inexperienced X and Y modellers respectively. As can be seen, the inexperienced method 'Y' modellers tended to make proportionally many more changes than the expert modeller. Excluding model 5 (which has an unusually high number of changes, as discussed in Chapter 5) the number of changes made by inexperienced 'X' modellers was again closer to that made by the expert.

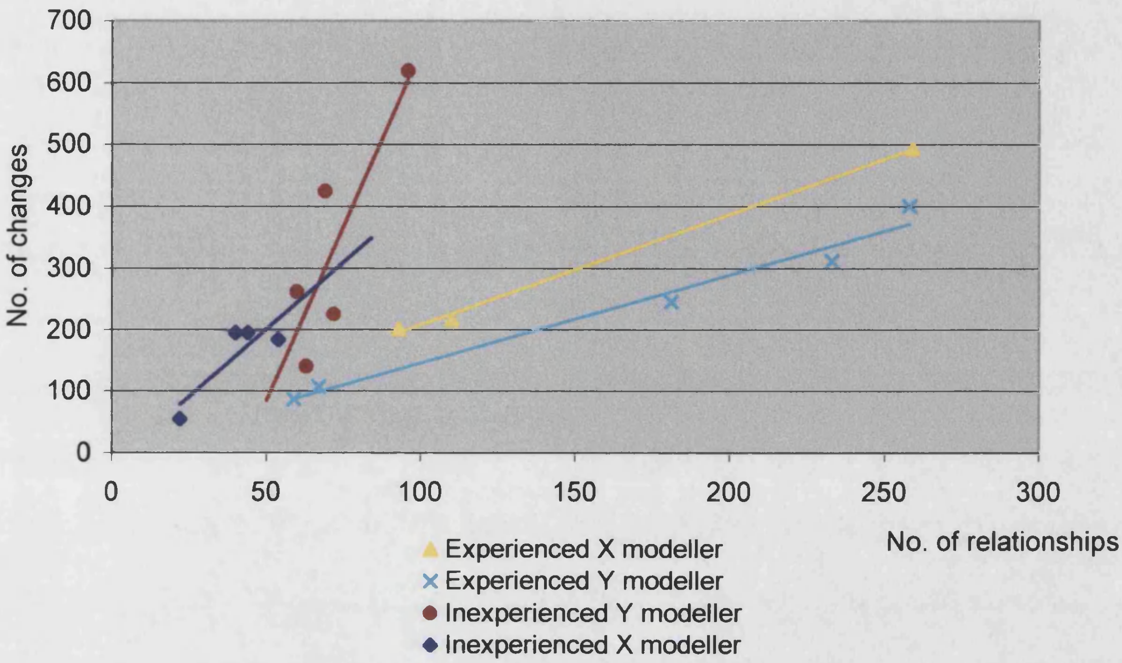


Figure 6.14 No. of relationships vs. no. of changes (excluding model 5)

We now turn to the specific strategies that were adopted to try to make modelling easier for inexperienced modellers.

Matching mental concepts

A modeller is probably more likely to include all of the relevant concepts in a model if they can be represented in an obvious and clear way. The observed levels of model completeness may therefore offer some guide to how well mental concepts were matched by modelling concepts. In the secondary study, the average completeness achieved by the modellers after brief training was 43%. This was quite close to the average completeness level achieved in the main study by inexpert method 'Y' modellers (35%) but falls far short of that achieved by inexpert modellers with method 'X' (86%). Although lacking in expertise, the modellers in question (computer science students) could be expected to have some aptitude for object modelling and had received more training in it than the other inexperienced modellers in this experiment. Hence the figure of 46% is perhaps surprisingly low. The secondary study therefore tends to support the method 'Y' completeness results for inexperienced modellers. Far greater variability was observed in the ultimate completeness of method 'Y' models, possibly indicating that completeness is more dependent on individual ability or prior knowledge when this technique is used.

Overall, there was a large disparity in completeness figures for the inexperienced modellers between the two techniques. Inexperienced method 'X' modellers, were probably able to express their mental concepts more easily in their models than inexperienced method 'Y' modellers. Why should this have been the case? Reviewing our psychological principles (Table 3.13), perhaps the principle that best encapsulates the idea of matching users' own mental concepts was 1.2 (*use conceptual models that match mental models*). The idea behind this principle was that users and modellers' mental energy can be conserved if the need for mental conversion is minimised. It is hard to assess our degree of success in meeting this objective since one cannot be privy to the modellers' own mental models. However, their behaviour offers some clues.

The less experienced 'X' modellers in this study seemed to have no difficulty with the most concrete concepts: *person, organisation, document, place and system.* They appeared to have a good intuitive grasp of what these meant. But the concept *physical object* was used rather indiscriminately, as a kind of catch-all, and the concepts *category* and *conceptual object* were used interchangeably. For example, the idea of a product type was quite likely to be modelled as a physical object (probably reflecting the fact that many products are physical objects) whereas it is more correctly modelled as a category. It seemed to be harder for users to grasp these more abstract concepts (principle 2.1 *model concrete concepts, not abstract concepts*). The concept *activity* was also slightly problematic. Perhaps it is not one that most people readily identify with, possibly being too general. Overall, the categories *physical object, category, conceptual object* and *activity* all seemed too general to be useful (principle 2.8 *model using concepts at an everyday level of generality*). It is possible that the ubiquitous idea of a software object has become so well-known that the term 'object' is now considered suitable to describe almost anything. If so it might make more sense to use a different term in method 'X'.

Only the expert modeller (J) defined generic concepts in models. The less experienced modellers defined only concepts that were pitched at an everyday level. But there was some apparent confusion about generality, with modellers apparently switching levels indiscriminately (one example was failing to distinguish between product and product type). This may primarily be a result of language since, in spoken and written English, it is quite acceptable to use the same word at varying levels of generality. For example, if asked to give examples of cars, people quite acceptably give answers like "Ford", or "Ford Fiesta". Oddly enough, few people would say "my car", or "the Ford Fiesta with registration number LGW 355F" which would be a more 'correct' answer from a conceptual modelling perspective. When people say things like "Ford" they obviously do not mean that the car manufacturer Ford is a car. They are probably envisaging a prototype or exemplary car and, to label it, they pick a convenient description such as "Ford" or "Ford Fiesta". Perhaps a problem with conceptual modelling is that it expects people to use language in an unnaturally specific way, making distinctions between things and their types beyond the everyday level of generality.

Analysis of model contents shows that both sets of less experienced modellers generally introduced concepts that existed only on their own spheres, as suggested by principle 2.3 (*introduce only concepts that exist in the user's world*) and principle 2.2 (*model business concepts, not technological concepts*). Method 'X' modellers seemed intuitively to regard model components as placeholders for familiar business entities, which may explain their difficulty with categories since these often did not correspond directly to clearly distinguished business concepts in the modellers' own worlds (the use of categories such as *gender* and *product type* can be viewed as a 'trick' employed by experienced designers to factor out or normalise common properties of more concrete entities such as people and products). Method 'Y' modellers had difficulty in grasping the concepts *class* and *association* beyond the most basic idea of a thing and connections between things. From the questionnaire responses, group members with prior experience of database design seemed to understand the idea of modelling better than those without, which seems reasonable since *object class* and *association* are essentially software design concepts (i.e. technological concepts) rather than business concepts.

Method 'X' modellers were restricted to a fixed set of categories (people, organisations, documents, etc.) and suffered less than method 'Y' modellers from the tendency to formulate ill-defined concepts that subsequently proved hard to refine. Modeller J observed that simply having the categories helped structure both thinking and discussions (principle 2.4 *allow models to be constructed from a rich extensible set of concepts*). The predefined concept set provided the important function of reminding method 'X' modellers to look for concepts of each type. Method 'Y' modellers had no predefined concept set, and often missed important aspects altogether (e.g. identifying people and organisations but failing to identify important activities).

According to one possible interpretation of the evidence in Appendix B, it is most probably optimistic to expect that mental concepts can be expressed in a direct way in conceptual models. There are significant differences in the ways mental concepts and model constructs must be expressed. If we accept the neural network view of brain function, there may be *no* mental concepts as such to express! Even if they exist, mental concepts are too loose to be easily formulated in the rigid way demanded by modelling techniques, and the impressionistic interpretation is probably the more natural mode of human thought. It is the expert, thinking with precision, who is unusual. Experiments have suggested that logic and causality are not innate modes of thought, and may simply be a 'syllogistic game': *"in highly-industrialised Western societies, people are trained to prove arguments about reality on the basis of representational propositions. In less industrial societies ... the form of proof is tied more directly to sensory impression"* (Solso 1998 p. 449). In other words, people do not necessarily think (and certainly do not argue) in a logical or stepwise fashion. It was clear in this experiment that the idea of a model as a series of logical propositions was foreign to several of the modellers. What we know of brain function must support this view since one would not expect a neural network to employ logic in developing inferences. The logic is constructed, only after conclusions have been reached intuitively, in the form of conscious rationalisation.

Perhaps method 'X' helped the inexperienced modellers more because it let them 'get away with' a looser interpretation style for much of the time. They had recourse to a formal interpretation (the English language view) when they needed it, that did not require them to exercise their own powers of logic. All method 'X' modellers made frequent use of the English-language interpretation, which clearly improved their understanding of the models (principle 1.3 *communicate effectively using the users' own language—in a structured, organised way*).

Inexperienced method 'Y' modellers simply failed to interpret their own models accurately. They were often surprised when the literal meaning of their models was pointed out. Some even objected, claiming that the literal interpretation was mistaken and that their own interpretation was the correct one (which, in one sense, it was). It appears that modellers need to switch between formal and intuitive modes of thought, using intuitive thinking for creation and discovery, but formal thinking when checking and correcting meaning. Hence it is better if they are not forced by the modelling technique into a continuously analytical mode of thinking.

It would have been interesting to follow through with principle 2.7 (*allow alternative concept definition methods and context-dependent classification*). Inexperienced modellers using both methods showed signs of treating their categories as fuzzy, interpreting them in different ways at different times. All modellers used unstructured or vague concepts at first, refining them until they had sufficient meaning. Method 'X' modellers were able to name concepts according to context but generally chose not to, presumably because to do so required additional effort and they felt it unnecessary. However, neither method supported genuinely fuzzy categories.

Understanding models

Inexperienced method 'X' modellers seemed intuitively to understand default pictures that were reasonably obvious, for the more concrete categories person, organisation, place, document, etc. (principle 1.1 *choose understandable and unambiguous representations—but retain formality if necessary,* principle 1.8 *use a rich set of recognisable 'lifelike' symbols—neither too general nor too specific*). However, they tended to be confused by more abstract symbols (e.g. that for *conceptual object*), either because the meaning of the symbols was not clearly distinguished or because no suitable pictures could be found (probably the former, since modeller J was able to find many useful images for abstract concepts). Modeller J went to some lengths to find suitable graphics to represent concepts and felt that this aided understanding (principle 2.5 *increase personal relevance by* *allowing users to choose their own symbols and by supporting alternative views*, principle 1.4 *maximise bandwidth using visuals where appropriate*). Some less experienced 'X' modellers also used graphics extensively, while others seemed to find the default graphics to be adequate. 'Y' modellers could not use graphics.

Modeller J stated that, with method 'X', the participants were *"intuitively ... more in touch with the picture—they are not faced with a big wiring diagram, they are digesting it in chunks"*. The modeller's choice of language is interesting here in light of the fact that he had not been exposed to any of the psychological thinking behind method 'X' and, on questioning, proved to be ignorant of psychological concepts such as chunking (principle 2.6 *support short-term memory using 'chunking' methods*).

Modeller J made extensive use of background images in method 'X' and felt this was of great value, presumably because it provided context and helped to distinguish different model areas from one another (principle 1.7 *assist recall and comprehension using context and other cues*). Neither technique allowed modellers to construct models that looked photographically like the corresponding business situations (principle 1.2 *use conceptual models that match mental models*). But, with the use of backgrounds and carefully chosen graphics, method 'X' models were visually reminiscent of their subject matter and this undoubtedly helped the modellers. Several method 'X' modellers also made good use of annotations, including text documents, presentations and images, all of which provided useful context in the form of background information.

Method 'X' modellers obviously enjoyed modelling with words and pictures, and seemed to construct and use concepts this way intuitively (principle 1.5, *maximise comprehension of model concepts by combining words with simplified pictures*). Modeller J reported that understanding was improved for group participants because of this. Using words and pictures together was not possible in method 'Y' and may partially explain why inexperienced modellers were observed to lose concentration and forget the meaning of their own model constructs on occasion.

The English-language interpretation in method 'X' can be obtained for a single concept, and at this level was undoubtedly useful to modellers since they were able to see a brief, uncluttered summary of the meaning of a concept without other information that might have confused them (principle 1.3). Method 'Y' modellers could have chosen to restrict information by redrafting subsections of their models but, in practice, did not (probably because it was unnecessary for small models and impractical for larger models). Inexperienced method 'Y' modellers clearly did suffer from a lack of focus, often attempting to read the model holistically, as if it had a meaning that could be stated in a single sentence—which was one reason why they misinterpreted their diagrams (principle 2.6).

It is unclear whether higher-level views could have provided more benefit (principle 2.6). Method 'X' offers a 'tree' view which is a higher-level view than the usual model view since it gives an overview of all components at the same time. However, modellers tended to use it infrequently, either because it was not useful or because they did not remember to use it (probably the former). Modeller J complained of the lack of high-level views in method 'X' and clearly did not regard the tree view as adequate in this regard. Method 'Y' does not offer high-level views.

Method 'X' modellers could use a different symbol for each concept, which allowed them to find and differentiate between concepts rapidly using preattentive processing (principle 1.9 *consistently use each symbol with one and only one business-related meaning*). The problem of having multiple representations for one concept did not arise with either method because each modeller used only one modelling technique at a time. Method 'Y' modellers used the same symbol for every concept (i.e. standard object modelling notation) so there was more sequential visual searching involved. However, the inexperienced method 'Y' modellers had no problem finding classes in the small models they produced. The expert modeller (J) who was adept at navigating large models, also experienced no difficulty in locating classes. But the models he produced were larger (see Appendix E) and would probably have been intimidating and difficult to navigate for the less inexperienced method 'Y' modellers without the ability to rely on preattentive processing.

The group members in organisation 8 judged their models to be less than complete. From an organisational perspective they were probably correct; no model could hope to fully match the complexity of the real organisation. But relative to the scope originally agreed, modeller J's models were essentially complete. The group members' assessment implies that they had a good grasp of the meaning of the models they were working with. Many analysts have experienced 'false agreement', when a user accepts a requirements statement or model as correct, when in fact they do not understand it.

The eventual correctness of models can also tell us something about how well the modeller understood models. To successfully refine a model until it is in a correct state, one must understand it well. The evidence is that all inexpert modellers tended to make a significant number of errors in the early stages of model development, so that the total error count in their models increased. Inexperienced method 'X' modellers eventually corrected their errors to produce models of reasonable quality. This implies that they must have been able to understand them. In contrast, method 'Y' modellers simply compounded their errors and several of their models then became disastrously incorrect. All failed to converge on a correct state. The poor performance of inexpert 'Y' modellers could have been due to not understanding the modelling technique, to practical problems in using the technique (e.g. reluctance to spend time in redrafting) or to lack of knowledge about the business domain. But it seems probable that it was, at least in part, due to their failure to understand the models.

Ultimately, the method 'X' models were over 80% correct for all but one model. This meant that they were probably good enough to be usable (e.g. as a basis for a database design). But most method 'Y' models produced by inexperienced modellers were less than 25% correct, making them unusable for practical purposes. In other words, it would have been quicker and easier to start again with these models than attempting to correct them.

One possible explanation for the difference in observed correctness is that we chose some very bad modellers to use object modelling and we had some rather gifted people using method 'X'. But this seems highly unlikely. If there is a simple conclusion that we can draw from the model analysis, it is that inexperienced modellers cannot be expected to interpret object model diagrams properly. Even if they understand the technique, they still may not possess sufficient mental energy or focused attention to read a model accurately. Instead they use a more impressionistic interpretation. Only truly expert modellers have a sufficiently-well developed ability to read models accurately. One may presume that, in an expert, the skill has become internalised so that little or no effort is required to translate a diagram into verifiable meaning. Even so, the sheer density of these models means that a lot of concentration is required and it is easy to slip up and miss an incorrect relationship, for instance, or a badly-named concept. Without external aids, people who are not expert do not have the expert's predefined mental structure and cannot emulate the expert's mode of thinking. An inability to understand models would not be a problem if the models were to be the subject of later, more formal quality control. But using seriously incorrect models to design databases and information systems can be seriously detrimental.

Making the modelling process easier

The large and obvious category buttons in the method 'X' tool encouraged the inexperienced modellers to start modelling quickly and in the right way. The inexperienced method 'X' modellers were already (at least minimally) familiar with the windows-icons GUI model and were able to begin to create models without the need for much additional knowledge (principle 3.10 *give users control and reduces unhelpful sets by avoiding jargon and making both the purpose and method of modelling intuitively obvious).* According to modeller J, the categories in method 'X' guided group discussion and prompted the group members (principle 3.3 *provide strong and clear model structure and frameworks to aid group comprehension and thinking.*). But inexperienced 'Y' modellers had trouble beginning their models—there was no obvious starting point for them.

The use of the English language interpretation tended to lend structure to the inexperienced method 'X' modellers' work. Typically they would model for a period and, after a while, view the interpretation to review what they had created. This normally prompted further corrective modelling or reminded the modeller of missing elements. Hence a naturally iterative cycle was set up that helped the modeller progress towards completing the model. Inexperienced method 'Y' modellers had no such guidance, other than the assistance provided by the experimenter. They tended to carry on modelling without pausing to review what they had done. When conducted using paper and pencil, method 'Y' can enforce no particular task structure, and most of the inexperienced modellers ultimately produced rather meaningless or notationally incorrect models.

Modeller J pointed out that method 'X' inherently restricts the view one has of a model, since one must explicitly open each component's window and there is a practical limit to the number of concurrently open windows. The restricted view forced inexperienced modellers to pay attention to a single concept at a time (principle 2.6 *support short-term memory using 'chunking' methods*). This seemed to help inexperienced modellers, presumably because their limited cognitive resources could be selectively focused on specific concepts, rather than being dissipated on the whole model at once. Method 'Y' modellers repeatedly tried to interpret several classes and relationships 'at once', which is not meaningful and is a quick route to cognitive overload.

But the restricted view also created the need for navigation, since modellers had to follow a series of windows to reach any given component. Modeller J felt that it was easier to know 'where you are' in method 'Y' models. But it is possible that traversing a series of windows was, in fact, beneficial for the less experienced modellers. It seemed to be an example of the spatial mnemonic in action (principle 1.6 *use layout effectively and consistently to increase understanding and improve recall*), a physical way for the modellers to remember where they had put concepts. As a physical memory, it was largely automatic.

Each method 'X' concept encapsulated its own properties and its relationships with other concepts. This seemed to be a useful form of chunking that made the modelling process easier, since modellers happily copied, pasted, moved and deleted components without concern for their contents or relationships. In method 'Y' this process is more messy because dangling relationships must be dealt with individually.

In method 'Y' inexperienced modellers did not understand the distinction between attributes and classes, and often modelled the same business concepts in both ways at the same time. No inexperienced 'Y' modeller grasped the connection between associations and attributes (pointers or foreign keys) and all produced contradictory models for this reason. In method 'X' these errors occurred infrequently, probably because duplication is visually obvious (principle 3.6 *tolerate, and reduce the likelihood of, inconsistency and other simple modelling errors*). In fact, method 'X' modellers can happily ignore any distinction between classes, attributes and relationships for most of the time (principle 2.9 *simplify modelling by reducing the number of modelling techniques and* *removing artificial or fine distinctions*). The only time that any kind of distinction must be drawn is when the decision is made to define a concept as text, date or number rather than person, organisation, document, etc. The decision can safely be deferred until the latest stages of modelling without any need for structural changes to the model.

Method 'X' offers the additional ability to locate components according to their type (people, organisations, documents, etc.) and this facility was used frequently by modellers. When creating a new component, in method 'X', one is forced to view the list of existing model concepts of the same type. This helps to prevent the creation of duplicated or alternative definitions of similar concepts, because it reminds the modeller of related concepts that exist already (principle 2.10 *store models such that memory can be supported by semantic associative retrieval*). Since method 'Y' does not incorporate the idea of concept types, modellers must check against the whole of a model to see if similar concepts already exist. Obviously this is harder to do, although in the small method 'Y' models produced by the less experienced modellers it is unlikely to have been an issue.

Method 'X' also allows images to be found by an associative method, and this proved to be popular, although the searching tool was rather slow. It is hard to see how the same richness of representation could have been produced by modeller J and some of the less experienced method 'X' modellers without this facility. Searching for suitable images would simply have taken too long. The experimenter's experience bears this out, since the 1500 images used by the image search tool had to be located and categorised in the first place, and this proved to be a very time-consuming activity, even using the Internet and CD-ROM image databases.

Modeller J reported major benefits from the visual aspects of method 'X' in running modelling sessions (principle 3.2 *use visual aids to support group exploration, learning, negotiation and refinement*). His 'Jazz' audio-visual demonstration model and extensive use of backgrounds and graphics illustrates his belief in these benefits (principle 3.9 *use all means available to maintain attention and increase arousal*). Method 'Y' sessions relied on paper copies of models and black and white overhead projector foils, which was inherently less fun for the participants as well as being a less flexible approach.

Modeller J was concerned that operating the method 'X' tool interfered with his ability to walk around in group sessions, and threatened to make the sessions less dynamic. A remote mouse was used, but when the modeller had to key information into the tool he was forced to return to the laptop. Remote keyboards are now available and may help by allowing the modeller to move around more. Alternatively, the modeller suggested nominating a 'scribe' in modelling sessions.

All modellers attempted to capture rough initial ideas before developing them, whether working in groups or individually (principle 3.5 *support brainstorming by facilitating capture of ideas and allowing easy model restructuring*). For method 'X' this process was eased by tool support and the modellers did not experience any significant penalty in undertaking what were sometimes quite major experiments with model structure (principle 3.7 *allow alternatives to be compared and explored before decisions are made*). For method 'Y' modellers, this meant redrafting and/or recasting the model several times as ideas matured. Redrafting was much more onerous in method 'Y' and discouraged the modellers. They also tended to make mistakes each time a model was redrawn (especially omissions).

Some aspects of group modelling that were discussed in Chapter 3 were not addressed in the experiment. Remote modelling (principle 3.1 *allow modelling 'at-a-distance' over time, by virtual groups*) was not attempted because of technological limitations in the modelling tool (see Chapter 7). Computer-supported collaborative modelling (principle 3.8 *allow models to be constructed jointly by smaller groups and individuals,* principle 3.4 *allow groups to work with model components already created and verified by other modellers*) was not tried because it was not feasible in any of the organisations that were ultimately involved in the study (it had been planned in organisation 8, but the eventual scope of the modelling exercise was too limited).

The modellers' productivity can give an indication of how well the modellers were supported during the modelling process. The results seem to bear out modeller J's belief that more was achieved with method 'X' in the modelling sessions than when using method 'Y'. Both experienced and inexperienced modellers were more productive with method 'X' than method 'Y', with an average difference in productivity of 147% for the expert modeller and 366% for inexperienced modellers. Because of the time spent making and fixing errors, method 'Y' modellers had to do more work to arrive at finished models than those using method 'X'. Using method 'Y' the productivity of inexperienced modellers fell significantly short of that of experienced modellers. But with method 'X' productivity was uniformly high for both experienced and inexperienced modellers.

The productivity results are intriguing and have no obvious cause. One possible contributory factor is the fact that the less experienced method 'X' modellers produced smaller models, on average. The number of potential relationships in a model increases with the square of the number of model components. Hence it would be reasonable to assume that larger models are disproportionately harder to work with and so take longer to correct per component. But it is hard to see how such a large difference in productivity (+366%) could result from a relatively negligible difference (-3.1%) in model size. Furthermore, it does not explain why the expert modeller's productivity was greater (+147%) using method 'X' despite the larger (+11.64%) models.

Modeller s using method 'X' made proportionately fewer changes to their models before arriving at a finished model (see Table 5.23). If we assume that each change to a model takes unit time to carry out then we could expect a linear relationship between volatility (number of changes per concept) and productivity, given constant model size. Hence one might predict that method 'Y' modellers would complete their models more slowly. However, the difference in productivity far outweighs any difference in volatility. Moreover, the average difference in volatility between the two methods is approximately the same for experienced (-18.8%) and inexperienced (-18.5%) modellers, whereas the difference in productivity (+366%) is much greater for inexperienced modellers.

For inexperienced modellers, there was a significant variance in average quality between methods 'X' and 'Y'. It is hard to explain such a large disparity. One possible cause is the relative difficulty of experimenting with method 'Y' models. The inexperienced modellers made many mistakes when they constructed models, using both techniques. It was important to be able to correct errors quickly and to try out different ways of modelling. With method 'X' it was relatively easy to change a model and the meaning (interpretation) of the model was always available in understandable terms. In addition, several method 'X' modellers used the automatically generated applications to discover if their models were sensible. Experimentation was therefore cheap and easy (principle 3.7 *allow alternatives to be compared and explored before decisions are made*). Consequently, inexperienced modellers using method 'X' were able to recover from their initial mistakes and, in most cases, went on to produce models of good quality. But inexperienced modellers using method 'Y' found it impractical to experiment with alternative model structures. Interpreting a method 'Y' model required deep concentration and a style of analytical thinking that was difficult for most of the inexperienced modellers. Generally, inexperienced modellers failed to interpret their method 'Y' models accurately and used a much more impressionistic approach. They tended to remain unaware of the errors in their models and consequently did not correct them. Some judged their models to be complete and stopped modelling even though significant problems remained. Inexperienced method 'Y' modellers also showed a reluctance to redraft models when problems surfaced, presumably because of the time and effort that would have been involved. Hence a sort of 'problem blindness' set in. The overall impression was that method 'Y' was simply too difficult or too much trouble for inexperienced modellers.

In contrast, the expert modeller (modeller J) consistently produced high-quality models with few detectable errors. Nine separate models were produced (four using one technique, five with the other) and each exhibited straightforward development and a consistent trend towards completion, despite the fact that the modeller was dealing with an unfamiliar business area. Modeller J was the only modeller who used both modelling techniques, and he expressed a preference for method 'X' over method 'Y'. But the results do not seem to support his view that quality was better with method 'X'. The method 'Y' and method 'X' models he produced were all judged to be 100% correct and complete. It is possible that the expert's own assessment of quality was sensitive to smaller differences than those measured in this study, and could be equated more with 'modelling style'. We corrected models only where large and obvious mistakes were present. A modelling expert can often find much to improve in a model that, in fact, represents its subject matter adequately. Our scale of quality was too coarse to register this kind of difference. Since the analysis of model contents did not reveal significant measurable differences in quality, Modeller J's preference could possibly be explained by the novelty of using the new technique or by the fact that he found modelling easier and quicker when using it.

6.3 Critique of experiment

The effectiveness and usability figures give some impression of the difference in performance between the two methods. However, they are rather speculative and should not to be taken too literally. Below we consider some of the factors that may have affected the reliability and validity of the research data.

Sampling model states

This analysis attempted to measure the total number of changes made to models. However, only a finite number of versions were analysed for each model. The versions selected for analysis were chosen to be representative of the models' states during modelling. Hence the analysis is based on a sample of model states rather than the set of all states that each model passed through. It is therefore possible that some changes will have been missed when consecutive model versions were compared. For example, if a component were renamed twice after saving one version of a model but before saving the next, then the two changes would be reported as a single change. A similar situation would occur if a component had been deleted but then reinstated between two consecutive model versions.

Completeness, correctness and error rate were unlikely to have been affected by the sampling of model components, since they tended to change slowly between model versions. However, the number of changes per component may have been affected and so the reliability of this measure is open to question. To test the sensitivity of this value to the number of model versions, some models were analysed using large numbers of versions (> 40). The results showed that the number of changes per component was increased, but only by a few percentage points (<10%). We can conclude that the reliability of the 'number of changes per component' measure is not significantly affected.

Sample size

Only 19 models were fully analysed as part of the primary study. The analysis of many more models could have provided more evidence of external reliability. But this was impractical in the present study because of the degree of effort involved in setting up the experiment and analysing models afterwards. Instead, the main defence against an unrepresentative sample was to use a secondary study for triangulation and to take care, when interpreting the results, to use all available information. The more in-depth, interpretative approach taken to a limited set of models arguably improves validity.

In the secondary study the results showed completeness figures of between 26% and 64%, with an overall average of 43%. This compares with an overall average of 35% (range 12% - 65%) in the main study for inexperienced method 'Y' modellers. One might expect the modellers in the secondary study to have had some prior aptitude for modelling, both because of their training and also because of the fact that they were computer science students. Therefore the results of the secondary study provide some reassurance that the completeness figures are in the right 'ball park' and offer some evidence of external reliability.

Modellers' domain knowledge

It is possible that some modellers were hindered by lack of domain knowledge. If this were the case then the validity of our measures would be compromised. In most cases, the modellers taking part in the study were already familiar with the relevant business area, having worked in it for months or years. However, modeller J was unfamiliar with the digital mobile telephony industry and new to the subject organisation. In the absence of relevant knowledge about a business area one can either try to gather information or attempt to model the business area anyway (which is perhaps less advisable). The expert modeller chose to gather as much information as possible so that, when the modelling sessions took place, he was in a reasonable position to understand what was said by group participants.

Model 5 was produced by the less experienced modeller I, who also was ignorant of the business area in question (investment banking, in this case). Her reaction was to start modelling. She attempted to gather information, but did so by consulting general texts that gave only non-specific information. As can be seen from the model profile, the result was an initial sharp increase in modelling errors, which were corrected only later when a reliable source of information became available (her contact in the investment bank).

We can conclude that bias may well have been introduced because of lack of domain knowledge for model 5, but was most probably not significant for any other models. The validity of our measures for these models is therefore probably not affected. For model 5, it would appear that modelling performance was depressed relative to what could have been expected.

Domain 'difficulty'

It would not make sense to compare modelling performance if some of the models in question were inherently more difficult than others. This would give a similar potential effect to a lack of domain knowledge: we could expect modelling performance to be depressed for the more difficult subject areas, threatening the validity of the measures for these models. However, rating domain difficulty is not straightforward. It can be argued that most business models are of roughly equal difficulty, since they tend to involve stereotypical situations. Business models typically refer to individuals or organisations who carry out business transactions such as ordering, checking, purchasing and supplying. The kinds of information recorded about people, organisations and business transactions tends to be quite predictable. Most of the models in this study involve scenarios of this type.

A further argument against the idea of inherent difficulty is that, at any given time, a modeller is either creating a single concept (such as a class or an attribute) or relating one concept to another (e.g. creating an association between two classes). Viewed in this way, all modelling reduces to a succession of similar actions, each of which is neither easier nor more difficult than any other. Provided that the modelling technique itself does not penalise one for having more concepts or more relationships, there is no reason why one model should be significantly more difficult than any other.

However, this argument ignores the problem of conceptualising 'difficult' subject areas. In this study the models that were arguably more difficult involved non-concrete or conceptual entities in non-trivial ways. It may be less obvious how concepts like these ought to be represented. For example, to model the structure of business policies or laws requires a greater degree of abstract thought by the modeller than, say, purchase order processing. Models 7, 11, 12 and 17 contained significant numbers of conceptual structures of this type. Of these, three were produced by the expert modeller and no discernible differences in performance were observed in comparison with more concrete models. The modeller clearly had no problem in knowing how to represent the concepts in these models. Hence we can conclude that no significant bias was introduced for these models and the validity of the measures is not compromised. However, model 7 (homeopathic remedies) was produced by an inexperienced modeller. The modeller experienced some difficulty in deciding how to represent the complex ways in which symptoms and treatments are classified. Significant bias probably was introduced because of the inherent difficulty of this model. Nevertheless, the numerical results do not suggest that the relative difficulty of the topic disadvantaged the modeller in any significant way. The number of errors was low and the productivity was high. The relatively small size of the model probably helped since it allowed the model to remain relatively understandable and manageable. The validity of our measures for model 7 is therefore probably not affected. If there is any effect for model 7, it would tend to cause modelling performance to be depressed in comparison with expected performance.

Modellers' prior training and ability

Modeller J had gained many years' experience in producing and reviewing method 'Y' models but had no experience at all in constructing method 'X' models. However, the skills were apparently transferable and he quickly became proficient in the new technique. There is no evidence that modeller J's learning curve affected the quality of the models he produced. Any bias due to differences in the expert modeller's ability between the two techniques was small.

However, the inexperienced modellers' knowledge and skill levels varied considerably. To minimise the impact of this variation, modellers were distributed between the two modelling techniques in an attempt to balance the mix of skills and knowledge. This seems to have been successful (see Chapter 5) but its effect is hard to quantify. Looking at the experimental results (e.g. Figure 6.2) it is obvious that there is a correlation between modelling performance generally and the modeller's experience level, and this applies to all measures and all modellers. In all cases, the least experienced tended on average to do worst and the most experienced to do best. The effect was clear for both modelling techniques, although it was stronger for method 'Y' than for method 'X'. It is possible to explain this difference between the techniques in terms of their relative levels of difficulty (see Section 6.3): method 'Y' was so difficult for inexperienced modellers that few managed to get beyond 'first base'. Clearly, therefore, some effect was at work here that was independent of modelling technique and we may reasonably attribute it to the differences between the various modellers, which seems to have had a major impact on modelling performance. What we cannot say is whether or not we managed to obtain an equal distribution of experience between the two techniques. We can say only that every effort was made to distribute the modellers evenly and that there did not *appear to be* significant bias.

The apparent difference in difficulty between the modelling techniques points to another potential source of bias connected with ability level. This is that the inexperienced method 'Y' modellers were given more assistance than those using method 'X'. It was difficult not to give greater assistance to the inexperienced method 'Y' modellers, since the absence of constraints allowed these modellers so great a degree of freedom that they often produced meaningless results (e.g. by drastically misusing the notation). As modelling progressed, most method 'Y' modellers remained unaware that their models were significantly incorrect and typically did not ask questions about their models or the modelling technique. In practice this meant that the experimenter had to take the initiative by pointing out significant errors in the models (but not demonstrating how to fix them), to ensure that the modellers were able to get started and go on to generate measurable results. With this source of bias it is likely that there *was* an effect, and that the effect was such as to elevate modelling performance beyond that which could otherwise have been expected.

Correction of models

Several of the models were corrected so as to provide a finished version for comparison. The method of correction, outlined in Chapter 4, is a potential source of bias, unconscious or otherwise. Every attempt was made to correct models in a fair and reasonable way, treating models in both techniques equally. This issue is discussed in depth in Section 4.1. However, whether it was successful or not cannot be proved. The level of bias introduced in the correction process could be tested by soliciting the opinion of independent experts who are familiar with the relevant modelling techniques. There are, as yet, no independent experts familiar with method 'X' so this option was impractical in the present study (see 'Further work' in Chapter 7). However, an alternative would have been to find a way to restate the models using a common notation, which would allow 'blind' and, therefore, unbiased correction.

In order to allow some test of validity and reliability, some models were informally corrected more than once to see if a corrected version could be produced with a different number of changes (Section 4.1). It was found that most corrections were very straightforward in nature and usually involved amendment or removal of incorrect relationships (see Table 6.2). Even inexperienced modellers generally managed to identify concepts well enough; it was in relating them to one another that problems tended to arise. Little scope was therefore available for correcting in alternative ways and, on balance, it seems reasonable to assume that the correction process was both valid and reliable. If there was an affect, it probably applied equally to all models that were corrected, regardless of modelling technique, since fundamental principles of modelling were involved that are independent of notation. This argument applies to the non-expert modellers (nine out of the ten modellers) but is irrelevant for the expert modeller, all of whose models were judged to be absolutely correct and complete according to the criteria used to judge the other modellers' efforts.

Hawthorne effect

In organisational studies it is always possible that observed performance improvements are the result of novelty and observation—the classic 'Hawthorne Effect' (Roethlisberger and Dickson 1939). Consequently, the cause of improvements may be mistakenly attributed. In this study it was certainly the case that modeller J became very enthusiastic about method 'X', to the extent of wanting to use it to the exclusion of method 'Y'. Possibly, therefore, the 147% productivity difference observed when he used method 'X' could be due in part to this effect. However, that would suggest that the modeller normally works at far below maximum productivity, which is possible but seems unlikely. In the case of inexpert modellers, novelty and observation may well have had an effect, but were probably not sources of significant bias since all of the modellers were doing something that was new to them. Therefore the effect, if any, could be expected to apply equally to all inexperienced modellers.

Summary

The various sources of error are summarised in Table 6.8.

Possible issue	**Issue**	**Likely effect**
Under-reporting changes	The no. of changes per component may be slightly depressed for models where fewer versions were analysed.	Probably no significant effect. If present, any effect would be random and would tend to depress the number of changes measured.
Small sample size	The small sample size reduces the credibility of the results.	It is difficult to be sure how representative the results are, but the fact that concrete patterns emerged and the secondary study result both tend to support the main results.
Variation in modellers' domain knowledge	The number of errors for certain models may be inflated as a result of the modeller's difficulty in gathering information.	The number of errors and error rate measured for model 5 (method 'X') are probably inflated. Completeness, correctness and productivity may also be reduced.
Variation in domain difficulty	Correctness for certain models may be depressed because the business area was abstract and hard to formulate.	The correctness for model 7 (method 'X') is likely to be depressed. Other measures may also be affected.
Variation in abilities of modellers	Modellers were chosen so as to balance skills and aptitude in each group.	Probably no significant effect. If present, any effect would probably be random and would tend to 'average out' between modelling techniques.
Variation in assistance given to modellers	All results for inexperienced method 'Y' modellers may be inflated because of additional assistance given to this group to sustain modelling.	Likely to cause results for less experienced 'Y' models to be inflated.
Bias introduced when models were corrected	A conscious effort was made to avoid bias (e.g. by making only the most minimal changes). Unconscious bias is possible, however.	Probably no significant effect. If present, any effect would probably be random and uniformly distributed between modelling techniques.
Hawthorne effect	It is possible that the expert modeller's productivity could have been inflated due to novelty (method 'X' only) and/or observation (both methods).	All measurements of productivity may be higher than one would normally expect.

Table 6.8 Summary of potential sources of bias and experimental precautions

Some of these (under-reporting changes, variation in abilities of modellers, bias introduced when models were corrected, Hawthorne effect) are judged likely to have had no effect or only small randomly-distributed effects. We need not be overly concerned with these sources of error since we are interested chiefly in comparing the two modelling techniques rather than measuring their properties on an absolute scale. Others sources of error (variation in modellers' domain knowledge, variation in domain difficulty) are likely to have had the effect of depressing the results for method 'X' models produced by inexperienced modellers (models 5 and 7). The remaining potential source of error (variation in assistance given to modellers) is likely to have had the effect of inflating the results for method 'Y' models produced by inexperienced modellers (models 2, 3, 8, 14 and 18). If there is any systemic effect, therefore, it will tend to affect only the results for inexperienced modellers, and it will tend to inflate the measurements of method 'Y' performance relative to method 'X' performance. Interpretation of the results must take this into account. No attempt has been made to adjust the data as measured in order to compensate for these potential sources of error.

6.4 Summary

This chapter has offered an in-depth interpretation of the experimental results presented in Chapter 5. The interpretation offers many insights. Chapter 7 takes up several of these issues and explores their implications. With regard to the psychological principles presented in Chapter 3, much evidence has been found for and against specific principles, and this evidence is analysed in detail in Section 6.2. The use of method 'X' improved productivity, which for inexperienced modellers was over 350% better than with method 'Y' (conventional object modelling). For inexperienced modellers the quality of the resulting models was also significantly better with method 'X' than with method 'Y': most of their method 'X' models were over 80% complete and correct. In contrast, most of the inexperienced modellers' method 'Y' models were less than 25% complete and correct (i.e. essentially unusable). For the expert modeller quality was reported as subjectively better, but our method of measurement was unable to detect the very subtle variations in the quality of the expert's work.

A number of innovations appear to have contributed to the improved performance with method 'X'. For example, the use of predefined concept categories seems to have helped focus thinking. Being reminded of existing model components encouraged modellers to reuse useful concept definitions. The English-language interpretation helped modellers to understand their models better and added structure to modelling efforts. Overall, we can say that the graphic view of model 'X' seems to be intuitively accessible to inexperienced users and modellers.

We found a strong correlation between the performance of modellers and their prior experience, as might be expected. For example, having created a database (e.g. using Microsoft Access) at some point in the past seems to be linked to a better understanding of the purpose and aims of conceptual modelling. But method 'X' helped apparently inexperienced modellers to emulate the actual performance of expert modellers. More than two-thirds of modelling actions by inexperienced modellers using method 'Y' were mistakes. With method 'X', this figure was reduced to less than one third. The productivity of inexperienced modellers was far below that of the expert modeller when using method 'Y'. But when inexperienced modellers used method 'X' their productivity was much greater, and comparable with that of the expert modeller.

We estimated the value of a composite measure 'effectiveness' for each modeller, for each model, using a formula based on the various underlying facets (completeness, correctness, error rate and productivity). A correlation of better than .9 was observed between these facets, indicating high internal reliability (Bryman 1989). By labelling certain effectiveness value ranges we are able to say that we found that the effectiveness of the expert modeller to be *poor* using method 'Y' but *very good* using method 'X'. The effectiveness of inexperienced modellers was *very poor* using method 'Y' but *good* using method 'X'.

It is possible to apportion parts of the total effectiveness score to different factors. In general terms, the contribution of prior ability seems to amount to approximately 36% of overall effectiveness, for all modellers. The contribution of method 'X' seems to amount to approximately 48% of total effectiveness, for all modellers. In other words, the choice of method has significantly greater impact on performance than whether or not one has prior experience or training. Since most of the modellers in this study who used method 'X' achieved model quality levels of at least 80%, we may ask whether 80% quality was acceptable (e.g. as a basis for information systems construction). This issue is taken up in the next chapter, along with several other relevant issues.

We also noted distinct patterns of model evolution. The typical 'expert' pattern is unaffected by choice of modelling technique. But inexpert modellers displayed a different pattern when using method 'X' from that when using method 'Y'. In several key respects, the pattern of inexperienced modellers when using method 'X' resembled that of the expert modeller, adding weight to the claim that method 'X' helps inexperienced modellers to emulate the performance of expert modellers. Finally, we looked at possible sources of bias and error in the experiment. Although various potential sources of significant error exist, none were considered likely to seriously distort the experimental findings. However, the numerical results should not be considered accurate measurements but instead serve merely as a basis for comparison.

7
Conclusions

The interpretation of experimental results presented in Chapter 6 raises a number of issues. This chapter summarises the major implications and suggests future research directions. One implication of this research is the prospect of untrained end users designing non-trivial business software applications. At present, many business end users can construct simple systems using desktop software products such as spreadsheets, but few are equipped to design more complex database applications. It is normally thought necessary when designing business applications software to employ the services of analysts and designers who can convert the user's view of the business into an abstract, system-oriented representation that can be used by system developers. The use of highly skilled analysts and designers is expensive and can be a bottleneck in the system development process. The need to express results using intermediate analysis and design notations introduces the possibility of miscommunication and can even alienate business end users. At present we cannot avoid the need to involve IT professionals in the production and maintenance of more complex business application systems. There would be many potential benefits if we could avoid this need.

7.1 Implications

Deskilling the modelling task

The results of this study suggest that, under the right circumstances, psychologically-inspired intervention can help novice modellers to be as effective as an expert modeller with many years' training and experience. The practical implications are far-reaching. We would see a significant reduction in the need for highly-trained and experienced IT professionals in requirements analysis and design tasks, at least for conceptual modelling in the 'narrow' sense (Chapter 1). Secondly, there would be a large and immediate increase in the number of people who could take on reasonably complex modelling tasks. This presents an intriguing prospect. The shortage of analysis and design skills has long been a barrier to quick and cheap business system development. Analysts take months or (more often) years to become effective, and inadequate and incorrect models are a prime source of poorly-designed systems, which are widespread. If the level of skill required to carry out modelling can be reduced sufficiently then highly skilled modellers may be required only to 'polish' models already formulated by end-users or less experienced IT specialists. In this way models could become cheaper and quicker to produce. Perhaps the most exciting implication is the prospect of cheaper and quicker development of business software, produced with less intervention by experts.

"Good enough" design: reducing design choice to empower end-users

Model-driven development, the idea of modelling an organisation as a way of creating systems, has a long history and well-understood benefits. But it can be argued that the software industry is moving away from this idea. There is strong competition to provide more features and greater programming capability in software construction tools. Great marketing appeal exists in technical selling points like object-orientation. Consequently, software vendors have made their development tools ever more sophisticated and comprehensive. For example, it has been claimed that client-server development is more complex and more time-consuming than development in earlier environments, given a constant level of application functionality. In client-server or Internet environments, non-trivial database applications may be created only by specialists who have background knowledge and a strong conceptual framework. Complex and powerful tools make it all too easy for inexperienced developers to create poor system architectures, which may not provide a good basis for continuing development. The large-scale development of integrated systems calls for experienced professionals who have been trained, perhaps to degree level, in programming and computer science, and who know how best to use the 'professional' tools wisely.

On the other hand, simple and limited systems may be developed with increasingly powerful end-user tools such as web page editors, macro languages, and spreadsheets. End users can employ these tools to create discrete systems for specific purposes. But without programming and design skills they cannot easily create integrated, interactive, corporate information systems in which large data resources are shared across the organisation.

Modelling techniques (e.g. UML) have likewise evolved away from any end-user bias. When conceptual modelling came into vogue during the 1970s and 1980s, the intention was first to represent the organisation and then to base software structures on that representation (Martin 1982). Now the emphasis is on representing software structures. It is no longer considered paramount to model the organisation in any direct sense (at least, in the object-oriented literature). Instead, one is expected to concentrate on software structures that may or may not reflect organisational reality (Rumbaugh et al 1991).

So this research suggests a way of developing information systems that is at odds with current thinking in the software industry. The present momentum is towards providing more flexible and more powerful programming tools such as object-oriented development environments. Our approach is the opposite: to make the tools simpler and more universal—by reducing their flexibility. The goal is to empower the end user. We achieve this goal, paradoxically, by *reducing design choice.*

Highly flexible but complex tools and development languages are replaced by simpler and more restricted (but 'good enough') design tools. The key distinction between this approach and the present-day use of 'user-friendly' products such as query languages and wizards is that we allow the user to create highly complex, professional quality results.

The spreadsheet is an example of how restricted functionality can increase capability. When using a spreadsheet to design an application, one is far less free than if using a conventional programming language such as C++. But, despite their inflexibility, spreadsheet packages revolutionised the development of financial information systems. One no longer had to program in the conventional sense, since a spreadsheet was a *good enough* design for a large proportion of modest financial applications. A similar level of capability can be gained by giving up flexibility in the creation of 'vanilla' business information systems. The flexibility that one must give up is the ability to meet highly specific and detailed software requirements or design standards. The capability that one gains is the ability to produce systems quickly that offer a close match with organisational reality. Systems generated directly from accurate business models may be *good enough* for a large proportion of modest business applications. One could therefore characterise this approach as the 'spreadsheet' for business system development. The major justification for this reduction in design choice is that a large part of what makes a system acceptable to its user is its degree of match with business reality (as subjectively perceived by the user). People find software usable if it works the way they do, given appropriately professional design standards. Hence software is easiest to use if it matches one's own mental concepts. It should be possible to deduce a usable system design from a statement of business concepts, using appropriate system design rules and 'usability heuristics' that mimic the expert designer's thought processes.

To realise this vision would require forms of modelling that are far easier to use than today's methods: object modelling and related techniques such as entity-relationship modelling. Ultimately, it may be possible to invent forms of modelling so transparent that no prior training is needed at all and very little cognitive effort is required. Modelling could become something you do *without thinking*. The results of this study do not suggest that this goal can be attained today. But perhaps the innovations introduced in this work have begun to show the way towards a reengineered form of modelling that could be effortless. Much thought has been devoted in the field of artificial intelligence to producing systems that understand users without being programmed explicitly (Green 1996). Any such system must be capable of modelling its user's world and adapting itself accordingly. It may be that the two advancing intellectual fronts—the effort to make modelling easier, and the effort to make modelling automatic—will meet somewhere in the middle. The result could be great benefits in cheap, available software.

Generating applications software

It has already been demonstrated that a large part of business system development can be automated (McKenna 1998) if appropriate conceptual models are available. Coupled with suitable application generators, modelling techniques like method 'X' could allow end-users to take responsibility for a larger share of the development of their own systems than is now possible. Table 7.1 presents this scenario alongside today's approach.

<table>
<tr><th>Type of system</th><th>Current approach</th><th>Possible alternative approach</th></tr>
<tr><td>A: Simple, isolated systems</td><td>Business people may construct relatively simple, isolated systems with end-user tools (e.g. using Lotus Notes or by cutting and pasting between Windows applications such as spreadsheets).</td><td rowspan="2">Business people may construct most business systems using simplified modelling tools, with automated code generation, based on their own business knowledge only.</td></tr>
<tr><td>B: Integrated, corporate systems</td><td>Complex, integrated systems must be constructed by software experts using a formalised software process and 'professional' languages/tools (e.g. Visual Basic, SQL, Java or C++).</td></tr>
</table>

Table 7.1 Possible alternative approach to system development

Today, there is something of a division between alternatives A and B. It is difficult to take a middle way. Systems constructed by software experts typically do not incorporate or link systems constructed by non-experts, because of incompatibilities between the technologies used or architectural inadequacies in the end-user systems. Under the possible alternative approach presented in Table 7.1, generated applications could presumably be made to follow corporate architectural standards automatically so that software experts would be able to customise and link them relatively easily into larger corporate systems. The alternative approach set out in Table 7.1 sounds suspiciously like the applications development golden age heralded by CASE vendors and IT industry pundits during the 1980s and early 1990s (Martin 1990), which has thus far failed to materialise. But it is similar to the 'template' approach that has yielded convincing benefits in software development (Hoffman and Rockart 1994). The logical structure and increased semantic content of method 'X' models make them a good basis for automated system development. Prototype tools have been developed (McDowell 1997, McKenna 1998, Visdeloup and Kocko 1999) that apply heuristic software design rules to create working applications on various platforms including Microsoft Visual Basic, Borland Delphi and HTML/Java, with corresponding database structures. The software generation tools are designed to generate functional client-server applications comparable to those produced by experienced software designers. The generated applications can be used either as prototypes or as finished systems. In our experiment this capability proved to be popular with some of the inexperienced modellers. Because generated applications are produced within seconds, they are useful *in the conceptual modelling process*, immediately showing the modeller how the model translates into a working software system. The structure of the generated applications is closely tied to the business concepts in the model, helping to make the application more easily understandable and maintainable.

Business process modelling and business process improvement

Our experiment focused on conceptual modelling in the narrow sense (Chapter 1). Method 'X' was applied in an object modelling style for the purposes of the experiment. But it is not solely an object modelling technique. The use of graphical components lends itself well to depiction of business processes. To demonstrate this, a prototype tool has been developed and field-tested (Santamaria 1999) that lets one model and animate the activities shown in method 'X' models. In accordance with psychological theory, each animated process is represented as a series of 'scenes' or a script. A common, but often unstated, task for requirements analysts is to analyse and restructure a faulty or inefficient business process before designing appropriate computer support. Method 'X' can be used for this task since process alternatives can quickly be modelled and viewed in a reasonably intuitive manner. The process animator uses the same model constructs and notation as the method 'X' modelling tool. The aim is to allow users to understand both data and process models more easily, because they use the same business concepts and look the same. Modellers do not have to learn separate techniques for data and process modelling since the process and data perspectives can be seen as alternative views of the same model.

Facilitating group work

The graphical representation of method 'X' can provide a basis for computer-supported cooperative work. Increasingly, organisations are becoming 'virtual' and geographically dispersed. But modelling (in JAD, for example) relies on the physical presence of a group of users and an analyst. Method 'X' can allow groups to participate in modelling sessions without physical proximity. This idea has been demonstrated in a prototype tool (Guy 1998), which allows models to be turned quickly into complete web sites that can be viewed using web browsers. In conjunction with videoconferencing or telephone conference calls this creates a useful means of modelling 'on the web'. The need for travel is reduced and users may easily browse models at their own pace. A wider audience may participate in modelling sessions, at a time that suits each individual (Ciborra 1996). On a more advanced level, a dynamic web modelling capability has also been designed (Grabner 1998, McConigley 1999) that will allow remote users to view models over the world-wide web, in real time, as they are modified by an analyst, simulating more closely how 'normal' modelling sessions are conducted.

7.2 Future directions

Verifying the results

The limited sample size in this research, coupled with an inability to control conditions, together mean that the study has not followed the classical scientific method in which hypotheses are tested through quantitative empirical research. A larger experiment, involving a greater number and variety of analysts, conducted under more strictly-controlled conditions, might be more acceptable as a scientific experiment and could help to verify the numerical results of this work and provide more scientific respectability. But, on the other hand, the research has validity in its own right as a quantitative study in which subjective observations have been interpreted and an argument developed about particular interpretations. From this perspective a case can be made for looking further at the participants in the experiment and their particular situations, with a view to understanding more deeply their motivations and the factors affecting their actions. It should be stressed that both views of the research—as an objective, quantitative experiment, and as a subjective, qualitative process of observation and interpretation—are themselves interpretations of what actually happened. Both are 'correct' and neither tells the whole story. But whichever view is taken, it is clear that further investigation will reveal more interesting insights.

Understanding why method 'X' appears to improve performance

This research has not looked very far into the question of exactly which factors contribute, and how much they contribute, to improved performance with method 'X'. It is apparent that some of the psychological innovations introduced in method 'X' provides benefit. However, this is not proven. The research has done no more than present a carefully-argued case for a number of changes, and then briefly test the result when a selected set of the changes is applied under specific circumstances. A fruitful avenue of research would therefore be to look into exactly which changes contribute most to performance. It could be, for example, that different factors affect different aspects of the modeller's effectiveness (such as productivity, error rate, completeness, etc.). Such an investigation would not be straightforward, however, since the modelling technique would have to be rethought if one or more of the aspects that have been introduced in this research were removed. One would have to very carefully construct alternative modelling techniques that work coherently. It would be difficult to rule out side-effects where the removal of one particular feature affects the way other features operate.

Testing method 'X' as a way to develop systems

As mentioned above, the method 'X' tool can be used to generate applications directly from models. It would be interesting to conduct further research, similar to the present study, in which this idea is tested by practical application. Central to this enquiry would be an investigation into the use of prototype applications as a guide during conceptual modelling (or, to look at it the other way, into the use of models as a way of constructing and refining prototype applications). The ultimate step in making a model more concrete—as suggested by psychology—could be to regard it simply as an expression of a required application. According to this view, the model would be equivalent to program code, and when executed would appear as a running (generated) application. The distinction between model and application would disappear in the mind of the modeller, since each would be an alternative view of the other (much as code written in C++, for example, can easily be confused with the resulting program). Whether it would be possible to construct models that simultaneously represent applications and organisations is unclear, however, and would itself be an avenue for further research.

Filling the quality gap: adding the remaining 20%

Probably the most interesting goal for further research is to find ways to help modellers produce even more complete and correct models. The results of this study show that novices using method 'X' can attain better than 80% completeness and correctness in less than half the time it takes an expert using object modelling. This compares to the almost immeasurably low quality of models produced by novices using object modelling. We need to find out how novices could produce 100% complete and correct models—without spending longer doing it. A necessary first stage would be careful examination of the specific errors made by novices. However, several other possibilities are discussed in the following sections.

Creating a method to support the modelling technique One way of improving quality might be to formulate a 'cookbook' procedure that modellers can follow when they use method 'X'. At present no such guidance is given. Despite its name, method 'X' is a notation and a tool but not a well-defined process. The fact that a modelling tool is used tends to push modellers in the right direction, but by no means guarantees that they will correctly identify all of the right components in their models, or that they will know how to relate them appropriately. Novice modellers make many mistakes before they produce correct models. The performance of the expert modeller with method 'X' suggests that he followed some kind of internalised method, conceptual framework or quality criteria to drive modelling and to judge completeness and correctness. Perhaps these rules could be identified and written down or codified in software, much as knowledge-based systems are designed to replicate an expert's knowledge. If this were possible then novice modellers might be able to emulate the expert's behaviour more closely.

Offering tool support for the method Formalising a method does not ensure that it will be applied or even understood. Many software development methods have been created, but none is able to guarantee quality. Therefore, producing a method alone is unlikely to ensure that better models are produced. There are several ways that a method could be supported and reinforced by the modelling tool. All involve a certain amount of 'suggestion'—of planting the right idea in the modeller's mind—since this is easier than having to correct a later, incorrect idea. The tool could incorporate "wizards" that assist modellers in capturing models by asking modellers questions that an experienced modeller would ask of him/herself. A wizard could also offer a simple, step-by-step, "fill-in-the blanks" approach for common business scenarios. This has already been demonstrated in a prototype tool (Johnson 1999) that prompts the user for details of activities with simple questions (who, what, where, when ...). Built-in help and computer-based training modules would offer general guidance and answers to common questions. A selection of good model examples would guide the modeller's thinking. The examples could include sample models for stereotypical situations and useful pre-built model fragments that can be incorporated into models. Finally, a stronger conceptual framework would help novices to emulate experts.

Providing a stronger conceptual framework Psychological findings suggest that novices can mimic expert performance if they are given the conceptual framework they lack for thinking about a problem. Method 'X' provides (and, in fact, enforces) a stronger conceptual framework than object modelling. It is quite possible that the improved performance by novices with method 'X' is at least partly due to this fact. Therefore we could consider ways of strengthening the conceptual framework offered by method 'X' to the novice modeller.

One idea is to provide a more prescriptive structure for models, so that concepts may be related only in specific ways. Currently, it is possible to link any business component to any other business concept. Whilst it would probably be inadvisable to completely exclude arbitrary links, it could be useful if the most likely connections were favoured in some way, so as to steer the modeller towards a structure that is more likely to be correct. Method 'X' models incorporate known types (person, document, activity, etc.) and so the idea of favouring likely links is eminently achievable. However, it would need to be based on a careful analysis (possibly statistical) of the connections in typical models. The use of fuzzy logic is being tried as one method of implementing this (Mooney 1998).

Taking the idea one step further, the definition of a set of roles for stereotypical situations could allow the possibility of prompting the user for likely missing information. For example, even at the most generic level, any activity is likely to involve components in specific roles (see Table 7.2). This approach could be extended to components of other types (although these would typically have fewer roles to fill). If more specific types of activity were listed then more specific roles could be named (e.g. if the activity were of type 'ordering' then specific roles might include customer, supplier, order, order fulfilment, invoicing, etc). These ideas are similar to the use in artificial intelligence of frames and slots. Some work in this direction has already been carried out with method 'X' (Santamaria 1999).

Role	**Played by components of type:**
Actors who cart out activities	Person, Organisation, System
Customer activity is carried out for	Person, Organisation
Inputs used or outputs produced by activity	Document, Concept, Physical Object
Location at which activity takes place	Organisation, Place
Time period in which activity takes place	Date/Time
Type of activity	Category
Sub-activities of activity	Activity
Event that triggers/is triggered by activity	Activity
Record of activity	Document, System

Table 7.2 Roles

Method 'X' with non-business systems

Method 'X' is designed to be useful in the analysis of business systems. To this end its modelling primitives are the concepts—people, organisations, documents, activities, and so on—that business systems contain. Models may thus be constructed from familiar concepts at the modeller's preferred, everyday, level of generality, in accordance with psychological theory. However, there is no reason why the modelling domain should be restricted to business systems. Every technical domain contains a primary set of concepts that could replace the set used in this experiment. This was seen in the present experiment, in the homeopathic medicine model where few of the concepts could be modelled using any of the modelling primitives except for 'category'. Implementing this idea would require an analysis of the commonly-occurring concepts in specific application domains. Moreover, it was found that some of the 'business' categories (e.g. conceptual object) were too generic to be easily used. So, even with business systems, it may be fruitful to formalise more specific base constructs.

Giving modellers an overview

Method 'X' allows the modeller to view a model in a variety of ways. A model can be examined in graphical form (icons and windows). Or it can be viewed and browsed in tree form. Alternatively, its

English language interpretation can be viewed (text form). However, the expert analyst pointed out that method 'X' lacks the ability to provide an overview of a model. Unlike object modelling, there is no model view that shows everything at once. This is intentional since the 'big picture' in traditional modelling techniques can often be confusing and off-putting. However, the absence of any overview means that it is possible to lose track of a model's contents and one's current 'location' in the model. Therefore it may be beneficial to construct an additional way of viewing method 'X' models, which in some way replicates the 'big picture' offered by object modelling. This would provide context and may possibly help to avoid redundant relationships. To this end work is proceeding on a prototype model viewer that allows models to be navigated in three-dimensional space (McMahon 1999).

Helping to avoid redundant relationships

By far the most frequent error made by all modellers is the creation of incorrect relationships. A large proportion of these incorrect relationships are redundant ones that could be deduced from other relationships. A prescriptive approach to the connections between components as outlined in Table 7.2 would go some way towards reducing the likelihood of these errors. But could more be done? One difficulty in identifying redundant links automatically is the fact that the modelling tool cannot tell whether a relationship that is potentially redundant is, in fact, required. For example, consider the situation where concept *employee* is related to concept *department* by two separate links. This could be an error if we know that both links are intended to represent *employee works in department*. But if we know that one of them represents *employee manages department* then there is no error.

Despite this there are various strategies for identifying potentially redundant relationships that could be used by a modelling tool to assist the modeller in weeding out unwanted relationships. For example, the tool could offer a view in which all implied relationships are shown (to a depth of say, two links) as 'ghost' components alongside other components. This would help make redundancy more obvious. Relationships most likely to be redundant would include those where the existing cardinality was identical to that of the implied component. Implied relationships that match existing relationships in name (i.e. role) are possibly redundant. At the most basic level, implied components that match explicit components in type (e.g. if an activity has two locations) are potentially redundant, although reporting all of them as errors could overload the user. To implement these ideas would require a careful analysis of the specific nature and causes of incorrect relationships. Fortunately, many such incorrect relationships may be found in the model versions that were analysed for the present study.

Building more intelligence into the technique

An alternative to helping modellers construct models more correctly is to deduce the intended meaning of a model more accurately (in other words, to permit a wider range of models to be 'correct'). Once again, the fact that method 'X' models are constituted from known types can help, since the statistical likelihood of any given pair of types being related in specific ways can be computed by inspection of known, correct models. Thus a knowledge base of probabilities could be built up, much as a human modeller gains experience by being exposed to models until he or she is able to recognise suspicious or incorrect structures at a glance, often without knowing precisely why. Neural network pattern recognition or fuzzy logic may be reasonable ways of implementing this. A statistical analysis of the models produced during this study has shown that certain cliché combinations do indeed occur repeatedly and hence a tool that used these to interpret models probabilistically (i.e. by guessing relationships) would be bound to do better than chance. There are problems in probabilistic interpretation, not least the inherent unpredictability that would result. However, experience in the present study has shown that modellers (even if experienced) are generally unable to predict how the modelling tool will deduce relationships, even though this is entirely deterministic and subject to a very simple rule. What they do instead is hope that the tool will get the relationship right and, if it does not, correct the model by explicitly stating the correct relationship. Hence we must try to ensure that the tool deduces correctly as often as possible, and rely on the modeller to correct the small number of instances where the deduction is wrong.

A second problem with probabilistic interpretation is that the tool will produce varying interpretations, from the same constructions, as the model evolves (because changes in one part of a model may affect the way another part is interpreted). This may result in unexpected and possibly unseen side-effects, much as a human analyst may make incorrect but unstated assumptions upon learning some new fact. In fact this already occurs to some extent because the tool already deduces some relationships, and experimentation will show whether it is enough of a problem to require corrective action.

Merging information systems development and use

The most recent 'lifecycles' for software development projects (e.g. Stapleton 1997) envisage software development as a process in which several simultaneous cycles coexist. This is a logical development from the early 'waterfall' and later 'spiral' methods (Wirfs-Brock et al 1990). It successfully captures the parallel nature of modern system development. But it still treats the development and use of information systems as separate activities. Notwithstanding the history and present constraints of information systems development, this is an odd way of approaching things. End-user computing and prototyping have taught us the value of working with designs to see how they pan out in practice. There is no reason why development and use should necessarily be separated other than today's practicalities to do with tools, skills and political control. The approach to information systems development suggested by method 'X' can be couched in terms of a sequence of activities, as shown in Figure 7.1.

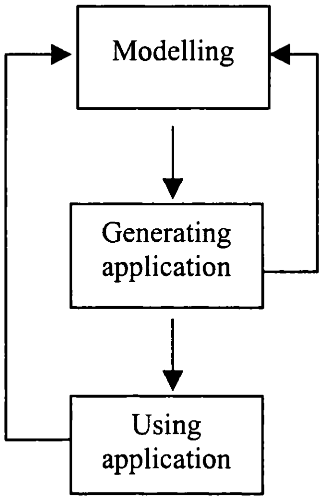


Figure 7.1 Modelling-generating-use cycle

A model and its generated application are closely related but distinct. Is this separation necessary? Could we envisage a situation where the model *is* the system, as far as the user/developer is concerned? It is reasonably easy to imagine that models in method 'X' or in a similar style could be used as interfaces for working applications. They are already in GUI form and therefore look a lot like applications designed for Microsoft Windows and Internet browser-compatible environments. The process of application generation in the present method 'X' tool creates completely standalone applications, that mimic the model's appearance to some extent but operate entirely independently of the modelling tool. To fully collapse application development and application use into a single process, this procedure would need to be modified so that the modelling tool acted as an application for whichever model is currently open. In other words, the model *would become the application.* To change one would be to change the other. This is already possible in small ways in tools such as Microsoft Access. There are of course many practicalities to be considered before this prospect could become a reality, however. The problem, for example, of what to do with data when existing applications are restructured would have to be addressed. But there is no reason why the existence of model-based applications should not be feasible, and it is perhaps surprising that suitable tools are not already commonplace that use this idea.

7.3 Conclusion

By applying psychological ideas to the conceptual modelling problem, this research has produced results that highlight some interesting possibilities. Using design techniques that take our knowledge of perception, memory and other psychology into account, it may be possible to produce higher-quality conceptual models with less training. It may be feasible for business end users to produce their own conceptual models even for more complex systems. It may be possible to produce prototype applications from method 'X'-style conceptual models, helping people to understand the implications of their models and speeding the design process. It may even be possible to create full applications directly from models, allowing business end users to produce and maintain quite sophisticated business application systems. These possibilities could have serious real-world implications.

This research has demonstrated that it is possible to measure in a practical way the quality of conceptual models and the performance of conceptual modellers. Contrary to the established view, conceptual modelling is not a black art but a relatively predictable activity where the outcome depends largely on known factors. Choice of modelling technique is the governing factor and the modeller's prior experience has a lesser, but still important effect. For modellers who are not highly-trained experts, the choice of modelling technique can have a very serious impact on model quality (and therefore on design quality), making the difference between results that are adequate and results that are essentially unusable. For modellers who are experts, choice of modelling technique can dramatically affect productivity. This finding must at least result in further investigation if we are not to continue to waste effort and talent with tools that make our work harder.

From an information systems perspective, the introduction of psychological theory raises an interesting possibility: the prospect of information systems people gaining a deeper understanding of the psychological principles that govern so many aspects of their work. Today, awareness amongst information systems specialists of psychology and its implications is poor. Psychology is simply not on the agenda except, perhaps, in research into human-computer interfaces. But psychology speaks of the working of the mind. It is a most important reference discipline for information systems. Much of mainstream information systems work *is* psychology, in one way or another, or would be, if only information systems specialists were aware that they were dealing with issues addressed by psychology. For example, information itself is subjective; to understand the nature of information one must understand perception, memory and mental models, if not cognition generally. While philosophical speculation about meaning and information are interesting, it is ultimately the real and specific mechanisms by which information is perceived and meaning created within the mind that can tell us most. Perception, memory, mental models, group dynamics and other areas of psychology have enormous implications for information systems specialists: they can affect how we approach and communicate with our clients, how we organise work, the methods we use to design systems and the design of those systems. They lie behind the ways that people use systems and even the ways that we think about systems. Psychology offers a very practical, understandable (and *teachable*) view of meaning, information and information systems. Large dividends would be paid if only a small part of the attention now devoted to technological and philosophical matters were given over to consideration of psychological issues in practice, research and education, as the work in this thesis demonstrates.

Appendix A
I.S. Methods and Conceptual Modelling Techniques

This appendix looks at a selection of information systems methods, highlighting aspects that are relevant to conceptual modelling. The purpose is to provide a basis for discussion of existing methods throughout the rest of the thesis, and therefore we consider conceptual modelling in the widest sense as defined in Chapter 1. Methods have been chosen for inclusion if they are either intended specifically for or have a substantial component devoted to requirements and systems analysis (see the reference life cycle in Table A.2). Methods for strategic planning or general organisational problem-solving have been included only if they can be applied to the development of information systems. Methods that are concerned solely with software design or implementation have not been included. A historical perspective has been achieved by including both old and new methods, highlighting the variety of approaches that have been taken to information systems analysis and design during recent decades.

A.1 Introduction

Classification of methods

Because of the enormous number of methods available, the scope of this review has been reduced by classifying methods into groups, which are given in Table A.1. Only one or two methods have been chosen for evaluation in each group. A reasonable balance has been attempted between methods that are widely used in practice and more theoretical methods. It is difficult to categorise methods satisfactorily in any simple way; the particular groups chosen are neither exhaustive nor mutually exclusive, and have been used solely for convenience in the absence of a more formal framework. They have been derived from several sources, including Lyytinen (1987), Davis (1987) and McGinnes (1993).

Evaluation categories

Each method has been considered under the following five headings, chosen to highlight particular topics of interest in this work.

Category	Examples of methods
Combined	Multiview (Avison,Wood-Harper, Vidgen and Wood 1996) SSADM (Bryant 1995, Ashworth and Slater 1993)
Critical factors	One Page Management (Khadem and Lorber 1998) Critical Success Factors (Rockart 1987)
Data-oriented	ORM (Halpin and Nijssen 1995) Information Engineering (Martin 1990, Finkelstein 1990)
Decision-oriented	Sprague and Watson (1996) Ackoff (1967) Simon (1957)
Normative	BICS (Zachman 1982, Kerner 1979) BIAIT (Carlson 1979)
Object-oriented	UML (Fowler and Scott 1997) Object-Oriented Analysis/Design (Coad and Yourdon 1991) Object-Oriented Modelling and Design (Rumbaugh et al 1991) Object-Oriented Analysis and Design (Martin and Odell 1992)
Problem-oriented	ISAC (Avison and Fitzgerald 1996, Lundeberg et al 1981)
Process-oriented	Modern Structured Analysis (Yourdon 1989) STRADIS (Avison and Fitzgerald 1996) Structured Design of Real-Time Systems (Ward 1986) ACM/PCM (Brodie and Silva 1982, Maddison 1983)
Prototyping	DSDM (Stapleton 1997) CASE method (Clegg and Barker 1994) Rapid Prototyping (Maude and Willis 1991) Prototyping (Vonk 1990, Boar 1984) Heuristic Development (Berrisford and Wetherbe 1979)
Socio-technical/ participative	ETHICS (Mumford 1996) Joint Application Design (Wood and Silver 1995, August 1991) Sociotechnical (Bijker and Law 1992, Pasmore 1988) STEPS (Floyd et al 1989)
Strategic planning	Information Strategy Planning (Martin 1990) Business Systems Planning (BSP) (Martin 1982, IBM 1981)
Systems approach	Information Systems Methodology (Wilson 1990) Soft Systems (Checkland and Scholes 1990, Checkland 1981) Viable Systems Model (Beer 1985)
Others	Speech Act (van Reijswoud and Mulder 1998, Lyytinen and Lehtinen, 1984) Boland and Day (1982)

Table A.1 Classification of methods

Extent of lifecycle addressed This section puts the method into broad perspective by listing its major steps and comparing them against the reference life cycle given in Table A.2. The purpose of this section is to place the method into overall context, to characterise it and to give an outline of its life cycle. The reference life cycle is a notional framework that allows discussion of real methods, some of which may deviate from it quite significantly. The reference life cycle mentions only *types* of work done in connection with information systems and can therefore be used irrespective of whether or not tasks are iterated, interleaved or overlapped. It also makes no mention of computers or other forms of technology and so is capable of being applied to information systems generally, regardless of the level of automation. Although some use of IT is often envisaged when an information systems method is used, this does not imply that models created during use of the method will necessarily be models of automated systems.

Requirements analysis	Identification of the need for information systems or changes to information systems, either by directly listing needs or through a process which seeks to arrive at requirements indirectly. Two separate planning activities are involved: strategy, where broad plans are formulated, and feasibility, where particular development proposals are evaluated. Requirements may then be expressed to greater or lesser levels of detail.
Systems analysis	Capturing and analysing detailed requirements for an information system, often by modelling the relevant parts of the organisation. (Note that the term 'systems analysis' is sometimes used in a rather wider sense than this; for instance, Miser and Quade (1985) use the term to refer to a much more general approach to organisational problem-solving.)
System design	Creating a model of a computer-based implementation which satisfies certain requirements.
Implementation/ maintenance	Constructing and supporting an information system.

Table A.2 Reference information systems life cycle

Steps that apply to modelling This section describes in more detail those parts of the method which are relevant to conceptual modelling, focusing on the techniques used and the ways in which people are involved in applying the techniques and using the results. The aim of this section is to determine whether the method pays much attention to this type of work (and, if so, what aspects). Some methods can be accused of paying insufficient attention to modelling or requirements analysis which could in turn lead to inadequacy in, or failure of, the resulting information systems.

Context and representation of model Focusing specifically on organisational modelling and communication, this section examines how any formal and informal conceptual models are represented, and how their representations connect with the wider organisation or problem domain.

The content of the models is summarised in the form of a 'meta-model'. The potential strength of the model representation as a basis for communication and negotiation of requirements is briefly discussed. The aim is to make plain the notations used to represent models, with the aim of assessing their usefulness as communication tools.

Approach to constructing model This section outlines the process(es) assumed by the method through which models are created, whether these are carried out in a straightforward, sequential, way or in a more complex process, perhaps involving iteration of steps or cross-checking of results. The expected roles of the participants in the modelling process (such as analysts and clients) and their assumed modes of interaction are outlined. The aim is to identify the procedures used to create models and to determine how the method assumes communication will occur between the major participants.

Assumptions Inherent in any method which seeks to provide a 'better' way of developing information systems is a set of assumptions about IT professionals, client organisations, the circumstances in which development takes place, problem situations, and so on. It is often difficult to identify assumptions upon which a method is based. Nevertheless, this section attempts to surface some of the more significant assumptions inherent in each method, with the aim in particular of bringing out unstated assumptions about applicability. The intention is to help place methods in context by identifying their assumed 'purpose'.

Figure A.1 illustrates how the categories given above are related to the components of each method.

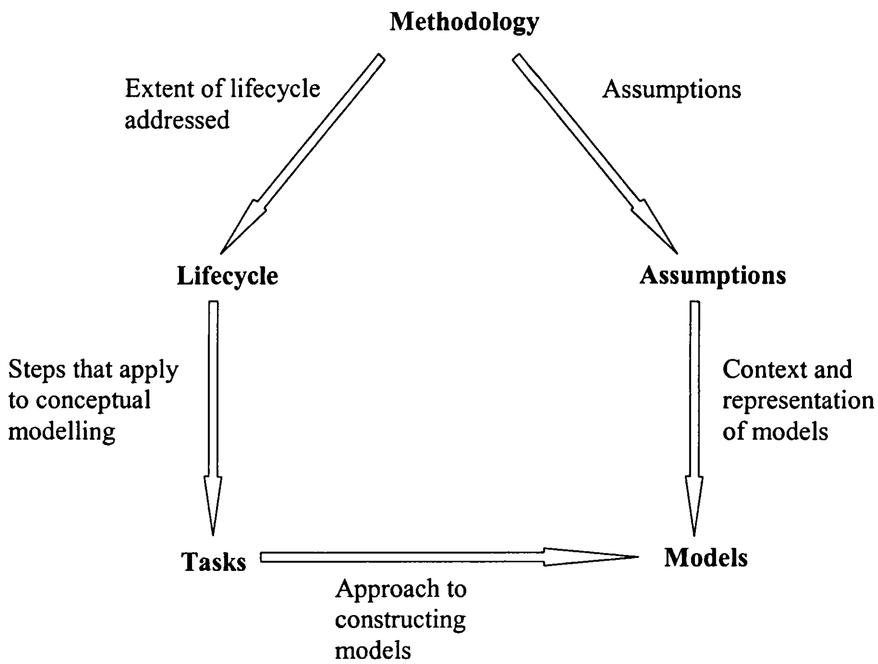


Figure A.1 Relationships between method components and evaluation categories

A.2 Combined methods

The diversity of methods for information systems development can be seen as evidence that no single method could possibly be complete in itself. In simple terms, it may be fruitful to combine the use of two or more methods so as to achieve better overall coverage. This is the strategy adopted in *Multiview,* which was developed primarily as a result of research into existing information systems development practices. Multiview combines techniques from several well-known methods.

Multiview (Avison, Wood-Harper, Vidgen and Wood 1996)

Extent of life cycle addressed Multiview includes components to cover the range of activities from planning to detailed design. Five stages are used: (a) analysis of human activity systems, (b) analysis of entities and functions, (c) analysis and design of the socio-technical system, (d) design of the human-computer interface, and (e) design of the technical subsystems.

Steps that apply to conceptual modelling The first four stages of Multiview are relevant to conceptual modelling. Stage (a) (analysis of human activity systems) corresponds to the activities described for Checkland's (1981) Soft Systems Methodology, used in this context as a planning technique. If the result of this stage involves new or changed information systems, the method proceeds to the next stage. Stage (b) (analysis of entities and functions) uses the root definition and conceptual model of the proposed system resulting from the previous stage. Data entities and functions in the proposed system are described in a modelling process similar to that carried out in SSADM (Ashworth and Goodland 1990). Stage (c) (analysis and design of the socio-technical system) is essentially a cut-down version of Mumford's (1983) ETHICS method. The results include a chosen socio-technical solution giving details of computer tasks and the roles and tasks of individuals. In stage (d) (design of the human-computer interface) decisions are made about the manner in which users will interact with the system. Prototyping (Maude and Willis 1991) is recommended as a way of developing a preferred user interface.

Context and representation of model One potential criticism of Multiview is that it produces no integrated model, relying instead on the models constructed at each stage. Some of the potential benefits of having a combined method are therefore lost. The modelling techniques employed at each stage are largely those described for each of the constituent methods, all of which are covered elsewhere in this appendix.

Approach to constructing model The five stages of Multiview are followed in a sequential fashion. Within each stage, model construction follows the pattern prescribed by the relevant method. Avison and Wood-Harper stress the importance of developing a holistic understanding of the problem situation as a result of carrying out each step, but no formal integrated model is constructed.

Assumptions The assumption in Multiview is that combining techniques from different methods will achieve comprehensive coverage of the systems development problem, and therefore better information systems. The aim of comprehensive coverage is certainly met, at least for requirements analysis and design, but it is unclear how well the various techniques can be integrated. For instance, there has been debate on the best way of linking Soft Systems Methodology with methods such as SSADM (e.g. Stowell 1992).

SSADM (Bryant 1995)

SSADM (Structured Systems Analysis and Design Methodology) is itself a combined approach, which was for a long period mandatory for all UK Government development of information systems. SSADM was designed by incorporating techniques from several other methods so as to realise the best features of those methods in a single approach, although its scope is more limited than that of that of Multiview.

Extent of life cycle addressed SSADM is aimed mainly at the analysis and design phases of the life cycle. Four main phases are specified (an optional feasibility study may be carried out first): (a) requirements analysis, (b) requirements specification (c) logical system specification, and (d) physical design. No support is provided for strategy or for construction, testing, production, maintenance or review. The SSADM approach is intended to be customised to suit the needs of each new project.

Steps that apply to conceptual modelling The first three phases are relevant to conceptual modelling. In phase (a) (requirements analysis) the existing environment is modelled through an investigation of the current functionality, data and system users. High-level requirements for the new system are assembled. Broad system options to meet the requirements are outlined and preferred options are selected. In phase (b) (requirements specification) the chosen business system option is analysed in detail. Data, process and behavioural models are completed and the objectives for the new system are made concrete. User roles in the new system are considered. Specifications are prepared as input to the design phases. Prototyping may be used to help refine the requirements. In phase (c) (logical system specification) technical options for the new system are defined and chosen options are selected and documented. Processes are specified in detail and user dialogues defined.

Context and representation of model SSADM makes use of several different graphical techniques to represent models. Dataflow diagrams, not unlike those used in Structured Analysis (Yourdon 1989), are used to depict process models for existing and required systems. Logical data structure diagrams, similar to the entity diagrams used in Information Engineering (Martin 1990) and Oracle's CASE*Method, are used to represent data models. Entity life history diagrams are used to represent the relationship between entity states and events external to the system (i.e. entity behaviour). Figure A.2 gives an example.

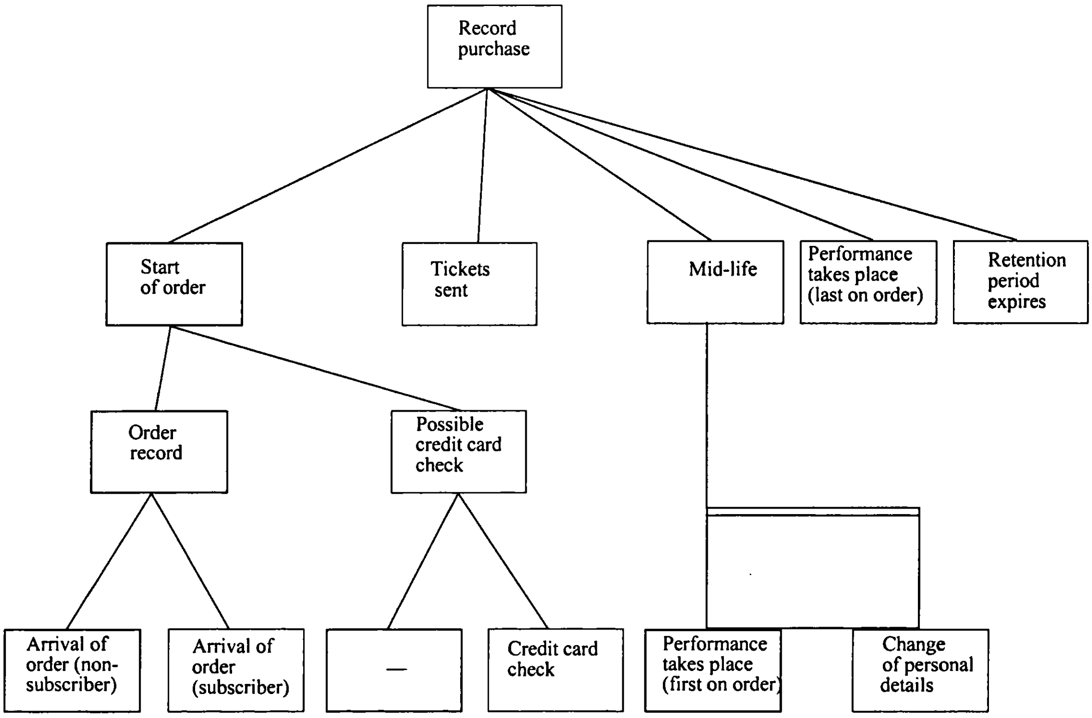


Figure A.2 SSADM entity life history diagram (part) (after Ashworth and Slater (1993) p.91)

A matrix representation is also used which summarises the life histories of data entities; Table A.3 gives an example.

Events	**Entities** (key: C = create, M = modify, D = delete)			
	Borrower	Video	Reservation	Loan
New customer joins club	C			
Customer leaves club	D		D	
New video put into stock		C	M	
Video taken out of stock		D		D
Customer reserves video			C	
Customer borrows video			D	C
Customer returns video				D

Table A.3 SSADM event/entity matrix

Approach to construction of model Ashworth and Slater state that there are no direct dependencies between the modules of SSADM version 4, and therefore a considerable degree of flexibility is allowed in the construction of models. However, it is generally assumed that a high degree of cross-checking will be carried out between models. For instance, the results of logical data modelling and relational data analysis are combined to achieve a final data model.

Assumptions In general, SSADM is a prescriptive and highly ordered approach to information systems development, although as stated above some flexibility has been added. SSADM relies heavily on structured modelling techniques, such as data flow diagrams and logical data structure diagrams, and places less emphasis on informally expressed requirements that do not fit into the process-data structure (although ad hoc requirements can be documented in a requirements catalogue). User interface specification is considered to be part of design rather than requirements analysis.

A.3 Critical factors approach

Critical success factor (CSF) analysis (Rockart 1987) is a simple and effective approach to defining the information needs of managers in an organisation. Although initially proposed as a method for top-level management (that is, for the definition of executive information systems) it has been applied successfully at all levels. CSF analysis is used in the *One-Page Management* approach to create an integrated performance monitoring system for the whole organisation.

One-Page Management (Khadem and Lorber 1998)

Extent of life cycle addressed Apart from its obvious uses in helping to define and structure organisational responsibilities, goals, targets, and so on, in the context of information systems CSF analysis is primarily useful as a requirements definition tool. In this role it can be used in two main ways: (i) to identify the organisation's goals and objectives, which can serve as input to a process of strategic information planning; (ii) to identify in detail the information required to monitor performance against goals, whether for an individual or for an organisation. The essential CSF analysis approach consists of three steps: (a) defining goals for each manager, (b) identifying critical success factors which indicate progress towards goals, and (c) deciding on appropriate ways of measuring performance for the critical factors. The resulting CSFs then form the basis of an information system, which may already exist in part or in full. The data required to report on some of the CSFs may be available immediately, while reporting mechanisms must be put into place for others. An automated information system can be built to report on the required information; however, this is not essential.

Steps which apply to conceptual modelling Each of the three steps is relevant to conceptual modelling. In step (a) (defining goals) the manager's own objectives are identified. It is expected that each manager's objectives will be compatible with those of the organisation, but there is no reason why they should be exactly the same since (in theory) each manager has his or her own role to perform. In step (b) (obtain critical success factors) suitable indicators of performance are identified for each goal. The indicators must be measurable and must reflect the manager's own performance, rather than that of his or her subordinates. In general, many candidate success factors will be identifiable for each individual, and it is necessary to choose a small number which are representative of the individual's broad responsibilities. Ideally, only one manager in the organisation will be responsible for any given CSF. However, in practice it is often the case that several managers claim responsibility for the same CSF, which should indicate to the analyst that there is cause for concern. From the CSFs identified for each goal, a suitable one is chosen for inclusion in the final system. In step (c) (define CSF measures and reporting frequency) a measure is given for each CSF which will indicate success or failure as accurately as possible. In addition, a reasonable reporting period for each CSF is chosen. The choice of reporting period is important because it determines the sensitivity of monitoring; longer reporting periods are less sensitive to short-term fluctuations in performance.

Context and representation of model Figure A.3 illustrates a general 'meta-model' for CSF analysis. In One-Page Management the goals, critical success factors and measures are expressed in report form, as shown in the example of a 'focus report' in Figure A.4. Employee performance is measured by comparing the level of achievement with goal levels for each factor. Minimum, satisfactory and outstanding goal levels define achievement targets that the individual can attain. The 'status' column shows current performance. The trend ('bad' or 'good') indicates whether the individual is getting better or worse relative to the last time the factors were reported on.

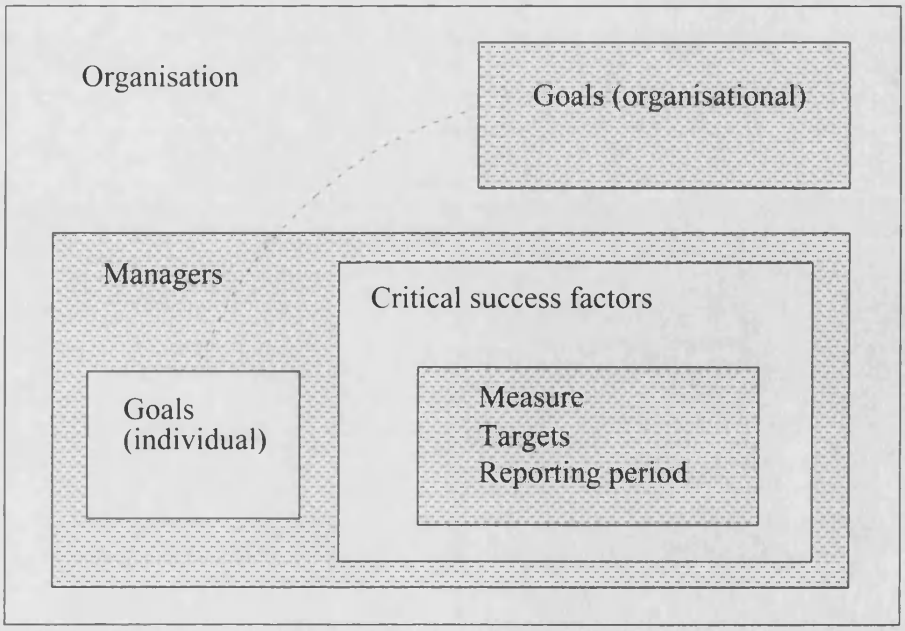


Figure A.3 Meta-model for CSF analysis

Summary of critical success factors

Employee: J Scott (Branch Manager)
Week ending: 1st August

Critical success factor	**Status**	**M**	**S**	**O**	**Trend**
		-------- **Goal Levels** --------			
New proposal revenue	$15K	$20K	$100K	$500K	Bad
Profit growth	-3%	0%	20%	50%	Good
Finalise staff targets	20 Aug	31 Oct	30 Sep	31 Aug	-

Factors on which employee's performance is measured

Current status of factor

Minimum goal level

Satisfactory goal level

Outstanding goal level

Figure A.4 One-Page Management focus report

Approach to constructing model Information about goals, critical success factors and measures is gathered in a straightforward manner following the three steps outlined above. It is claimed that the process can generally be accomplished within a few hours for each manager.

Assumptions Perhaps the key assumption underlying the CSF approach is that individuals are able to identify specific goals, for all aspects of their work, and that performance against these goals can be measured and reported on within a reasonable time and in a cost-effective manner. The extent to which the idea of working to a target and performance monitoring is compatible with organisational culture depends on the organisations themselves, but clearly some organisations would be more at home with the CSF approach than others.

A.4 Data-oriented approach

Data-oriented methods emphasise the need to devote attention to understanding and modelling an organisation's data structures before information systems are designed. To some extent they came about as a reaction against the more process-oriented 'structured analysis' methods which became popular during the 1970s and 1980s. *Information Engineering*, originated by James Martin (1990) and Clive Finklestein (1990) during the 1980s, is probably the most widely used of the data-oriented methods.

Information Engineering (Martin 1990)

Extent of life cycle addressed Information Engineering aims to provide broad coverage of the whole information systems life cycle. This evolving method has been developed in slightly different directions by its two main authors. According to Martin (1990), Information Engineering offers a continuum of life-cycles based on a general framework and on a set of techniques. The most appropriate life-cycle and techniques can be chosen depending on the particular environment and tools available in the organisation. The method has seven stages: (a) information strategy planning, (b) business area analysis, (c) business systems design, (d) technical design, (e) construction, (f) transition, and (g) production.

Steps that apply to conceptual modelling The first two stages in particular are concerned with conceptual modelling. The main purpose of stage (a) (information strategy planning) is the partitioning of the organisation's data into subject areas for further analysis, and the production of a plan outlining how these subject areas will be analysed. This stage also involves an examination of the current situation, the identification of management needs and a preliminary analysis of data and functions of the organisation. Although Information Engineering is not a general-purpose business strategic planning method, one of the main inputs to the planning phase is the corporate strategic business plan which documents business goals and strategies. In stage (b) (business area analysis) each subject area identified in the information strategy plan is subjected to detailed analysis. Comprehensive data and function models are constructed using entity diagrams, function hierarchy diagrams and process dependency diagrams. Entity life histories are mapped and detailed logic of processes is specified. Models from different subject areas are combined where possible to allow checking and to synthesise a canonical model. Stage (c) (business systems design) and subsequent stages are concerned more heavily with design than requirements. A preliminary design is produced to meet the requirements expressed during previous stages, and the scope of the software and hardware parts of the new system are decided. The remaining stages (technical design, construction, transition and production) are concerned with building and implementing the software and hardware parts of the system. Prototyping may well be used during these stages as a way both of testing requirements already documented and of eliciting fresh information.

Context and representation of model Information Engineering makes full use of diagrams at every stage in an attempt to improve quality and to allow effective communication. Figure A.5 illustrates one of the most important diagram types used in the early stages, the entity diagram. This diagram shows the relationships between business entities.

Approach to construction of model Information Engineering stresses the need for a formal, engineering-style approach to information systems design, although it is less formal than the 'formal methods' such as Z and VDM (Woodcock and Loomes 1988). The 'continuum of methods' approach allows for various levels of formality, ranging from end-user development, via relatively informal development using prototyping as a means of gathering requirements, to reasonably formal development making use of structured modelling techniques. It is up to the method user to decide on the most appropriate 'path' through the method and to choose appropriate tools and techniques.

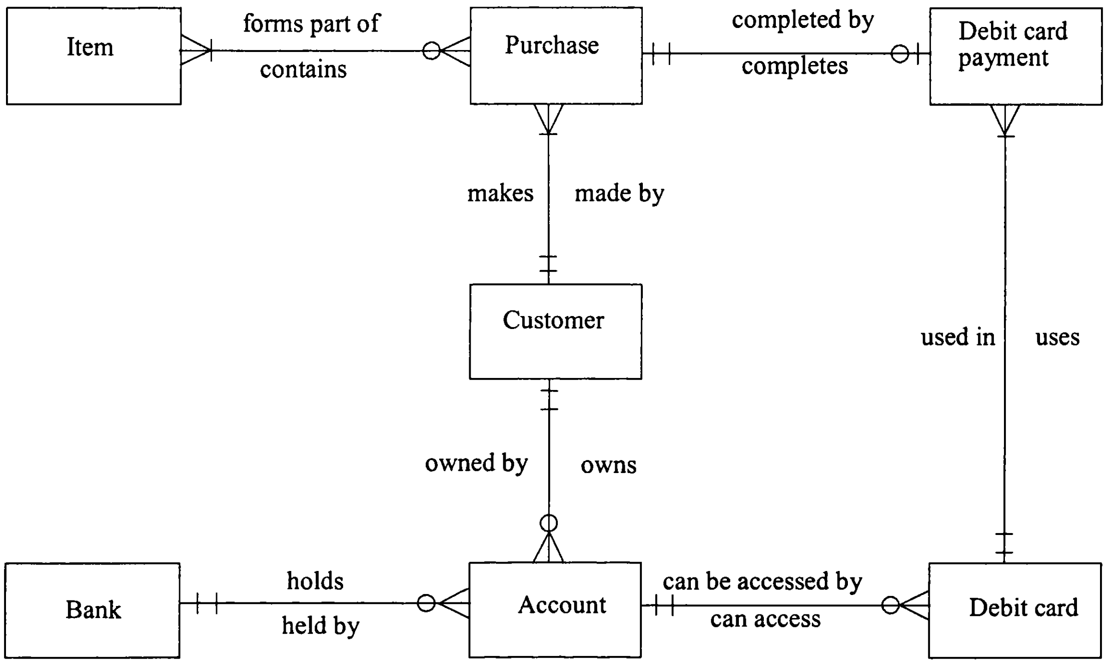


Figure A.5 Information Engineering entity diagram

Assumptions Information Engineering is based on the premise that it is more important to model information than any other aspect of an organisation (in the development of information systems). It is argued that concentrating on modelling processes—as done in most structured approaches—is tantamount to following a moving target, since procedures are considered to be less stable than data structures. A second, less explicit, assumption is that information systems can be designed successfully from largely deterministic models of an organisation's goals, data structures and processes, since Information Engineering devotes little explicit consideration to the concerns of individuals.

A.5 Decision analysis

In decision analysis-based methods the organisation is viewed primarily as a collection of managers who use information, in a largely rational way, to make decisions. Ackoff's approach to the design of management information systems was not a well-developed method per se but was typical of the class of decision analysis and decision support methods.

Decision Analysis (Ackoff 1967)

Extent of life cycle addressed Ackoff's approach involves five steps, broadly covering the whole life cycle: (a) analysis of the decision system; (b) an analysis of information requirements; (c) aggregation of decisions; (d) design of information processing; and (e) design of the control system. The emphasis is on the design of information systems, that are not necessarily wholly computerised, to support management decisions. Therefore, Ackoff has little to say about design issues specific to any computerised portion of the resulting information system. Ackoff's approach was designed to counter what were some prevalent but unfounded assumptions about the purpose and use of management information. These are listed in Table A.4.

Assumption	Ackoff's view
The critical deficiency under which most managers operate is the lack of relevant information.	An over-abundance of information is more of a problem.
The manager needs the information he/she wants.	Managers often have only a hazy view of what information they need to make any given decision, even if they can predict what decisions are going to be made.
If a manager has the information he/she needs then his/her decision-making will improve.	Managers may not draw the correct conclusions from data and may need assistance in analysing it.
Better communication between managers improves organisational performance.	Information flow between departments can lead to instability in the organisation as a whole.
A manager does not have to understand how his/her information system works, only how to use it.	Management must understand how their information systems work (several reasons given).

Table A.4 Unfounded assumptions about information cited by Ackoff

Steps that apply to conceptual modelling Every stage of Ackoff's approach could have some relevance to conceptual modelling. During stage (a) (analysis of the decision system) each type of management decision is analysed in detail. In stage (b) (analysis of information requirements) three classes of decision are considered: (i) those for which optimal solutions can be found and for which the information system can therefore compute optimal solutions; (ii) those for which models can be constructed but no optimal solution can be extracted, so that only heuristic or search procedures can be provided to aid decision-making; (iii) those for which adequate models cannot be constructed, so that some research (or guesswork) is required to determine what information is relevant. In stage (c) (aggregation of decisions) decisions with the same or overlapping information requirements should be grouped together as a single manager's task. For stages (d) (design of information processing) and (e) (design of control of the control system) Ackoff does not specify any particular design method, except to recommend extensive use of exception reporting. Ackoff also observes that the system should be flexible and adaptive, detecting and correcting its own deficiencies; this normally requires a system which is not wholly computerised.

Context and representation of model No meta-model is explicitly presented, but an implicit model can be inferred; this is shown in Figure A.6. The meta-model sees the business environment as a 'decision system' in which managers make decisions of varying types based on information provided to them by an information system. The information system may be partially computerised but is unlikely to be wholly computerised. Ackoff does not stipulate any particular representation for models constructed using his approach; we can probably assume that decisions, information requirements and so on would be represented in prose form.

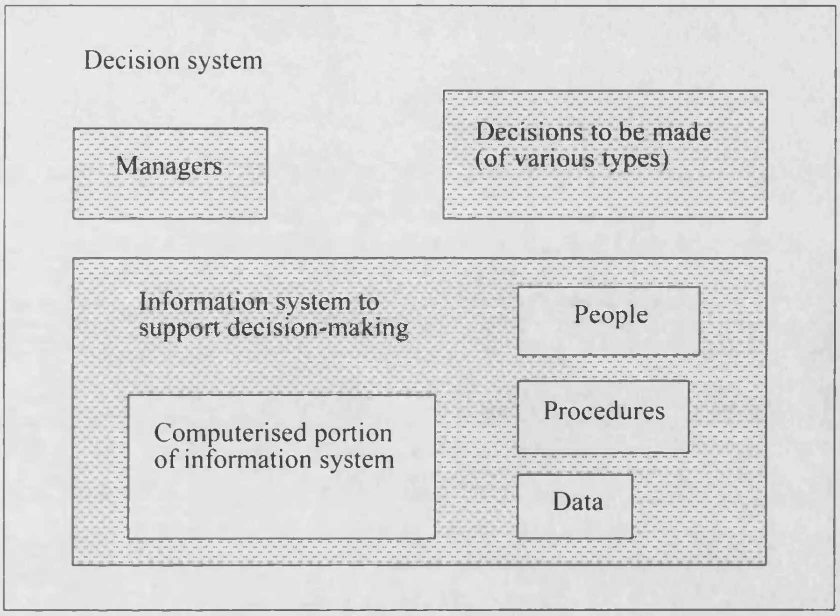


Figure A.6 Meta-model for Ackoff's approach

Approach to constructing model The model is constructed in a top-down fashion following Ackoff's five steps. An analysis is first made of all of the decisions which a manager or managers are likely to need to make and the information requirements of those decisions are defined, leading to the specification of an information system to provide this information. Some rationalisation of decision-making responsibilities may result from the initial analysis of decisions. Any computer support for the information system is then designed.

Assumptions Perhaps the most important assumption underlying Ackoff's approach is that information systems (or 'MIS' as they were more generally called at the time) exist primarily to support managerial decision-making. This limited view deals with only one aspect of today's information systems.

A.6 Normative analysis

Normative approaches seek to adapt standard solutions to the needs of particular organisations. The *Business Information Control Study* (BICS) method is based on commonality in the information requirements of different organisations.

BICS (Kerner 1979)

Extent of life cycle addressed BICS is a planning method which is used to identify information requirements at a high level, and hence provides input to the earlier stages of information systems requirements analysis. High-level data analysis is carried out. The steps (paraphrased) are: (a) identification of order types; (b) BIAIT analysis of order types; (c) selection of data categories; (d) Mapping data categories to organisational structure.

Steps that apply to conceptual modelling All of the stages of BICS are relevant to conceptual modelling. In stage (a) (identification of order types) the various types of order received by the business are identified. In stage (b) (BIAIT analysis of order types) seven questions are asked for each type of order (i) Are invoices created or not? (ii) Are orders recorded or not? (iii) Are sales recorded or not? (iv) Is the price negotiable or fixed? (v) Does the customer hire or buy the product? (vi) Does the supplier record the product destination or not? (vii) Is the product made to specifications or supplied from stock? The answers to each question are likely to vary between the different types of order considered. In stage (c) (selection of data categories) standard information categories called data inventories are selected according to the profile for each type of order. For instance, if invoices are created, then credit control information must be kept. The data inventories are (i) employee, (ii) facility (buildings, etc), (iii) vendor, (iv) money, (v) outgoing order, (vi) activity (work performed by people or machines), (vii) product, (viii) product description, (ix) customer, (x) incoming order, (xi) process description (relating product to work activity), and (xii) track (relating product to customer). In addition, standard information categories are included that do not depend on the way orders are processed (e.g. personnel data). Data classes within each applicable data inventory are then mapped onto the organisational structure in stage (d).

Context and representation of model The mapping of data classes to organisational structure serves as an architecture which BICS uses for further analysis (Zachman 1982). The results of this mapping are expressed in the form of an 'organisation versus data class matrix' as illustrated in Figure A.7. This matrix shows the roles that different organisational units play with respect to different data classes. For each data class BICS distinguishes between planned and actual data, and between value data and descriptive data. A statement of business problems and measures (key business indicators) is also produced.

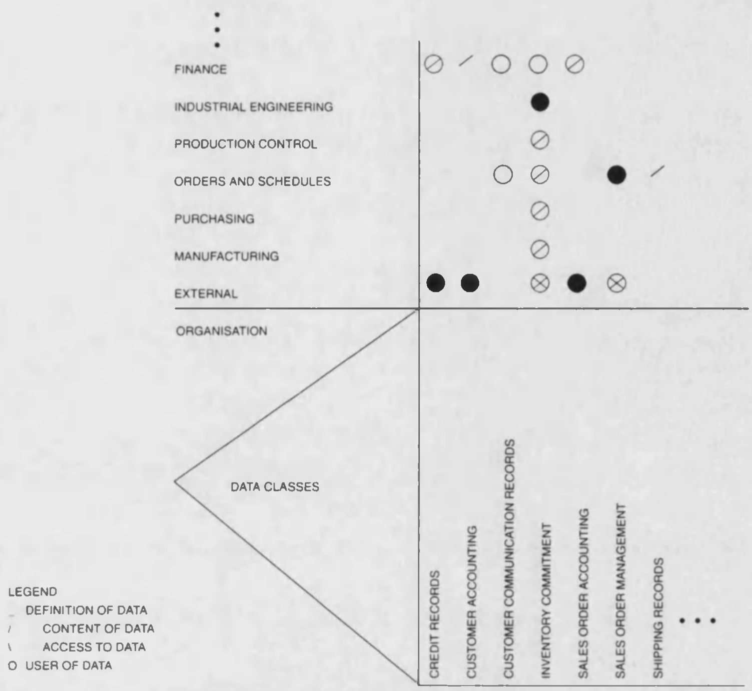


Figure A.7 BICS organisation versus data class matrix (after Zachman (1982) p.41)

Approach to constructing model BICS is essentially a top-down planning method which deals with classes of data rather than specific data entities. Its primary thrust is for a quick and cheap solution rather than extended information analysis.

Assumptions As a planning approach that aims to take advantage of commonality between organisations, BICS relies heavily on its assumption that the set of information handling procedures required by an organisation can be derived largely from the answers to the seven questions given above.

A.7 Object-oriented methods

When it came into vogue during the early 1990s, object-oriented design represented a new way of looking at software. Instead of the traditional split between programs and databases, software could be organised in terms of 'objects' containing elements of both program and data. Objects communicate with one another solely by passing structured messages, allowing data to be accessed and functions to be invoked. To match the new way of designing software, object-oriented analysis methods were developed. According to Coad and Yourdon (1991), *Object-Oriented Analysis* (OOA) was a way of merging the previously separate data and process-oriented approaches.

Object-Oriented Analysis (Coad and Yourdon 1991)

Extent of life cycle addressed It is difficult to place OOA in the reference life cycle. Although it is described as an analysis approach, OOA has much in common with design methods, and it is stated that a prior requirements analysis exercise may well be carried out. It seems safe to say that OOA is aimed at the detailed analysis and early design phases. The method used to derive the OOA model is a top-down approach consisting of five stages: (a) finding class-&-objects; (b) identifying structures; (c) identifying subjects; (d) defining attributes; (e) defining services.

Steps that apply to conceptual modelling In stage (a) (finding class-&-objects) classes are identified which the system keeps information about, interacts with, or both. Potential classes may be derived from structures, other systems (whether connected to the system or simply known to it), devices, things or events remembered, roles played, operational procedures, sites and organisational units. In stage (b) (identifying structures) two types of relationship between classes are identified. Generalisation-specialisation structures reflect distinctions between similar classes. Multiple generalisations are permitted, forming a lattice rather than a hierarchy. Whole-part (aggregation) structures show the components of objects, and may reflect relationships between an assembly and its parts, a container and its contents, or a collection and its members. In stage (c) (identifying subjects) subjects are chosen. These are high-level subdivisions of the problem domain and are useful in helping the reader navigate larger models and as a way of structuring the problem space to aid in modelling. They should be chosen in such a way as to minimise inter-subject dependencies and interactions. A subject may contain other subjects.

In OOA the storage of data is represented using attributes, which are identified during stage (d) (defining attributes). Attributes generally refer to atomic data values (although Coad and Yourdon also permit simple aggregate data items such as 'address'). Instance connections or associations between objects are recorded, and many-to-many relationships are permissible. In stage (e) (defining services) operations are identified and allocated to classes. Two types of service are identified: algorithmically simple (create, connect, access, release) and algorithmically complex (calculate, monitor). Only services of the second type are modelled, using service charts (similar to program flowcharts or pseudocode). A services/states table can be produced for each object showing which services are possible in which states. A very simple state diagram can also be constructed for each object.

Context and representation of model The OOA model itself consists of several 'layers': the subject layer (general description of the problem domain); the class and object layer (depiction of individual classes and objects); the structure layer (linking objects and classes using relationships); the attribute layer (adding attributes to object classes) and the service layer (adding service names to object classes, specifying their implementation and depicting message connections to show the invocation of services). Large models may be partitioned into more manageable subject areas. Figure A.8 illustrates an OOA object model.

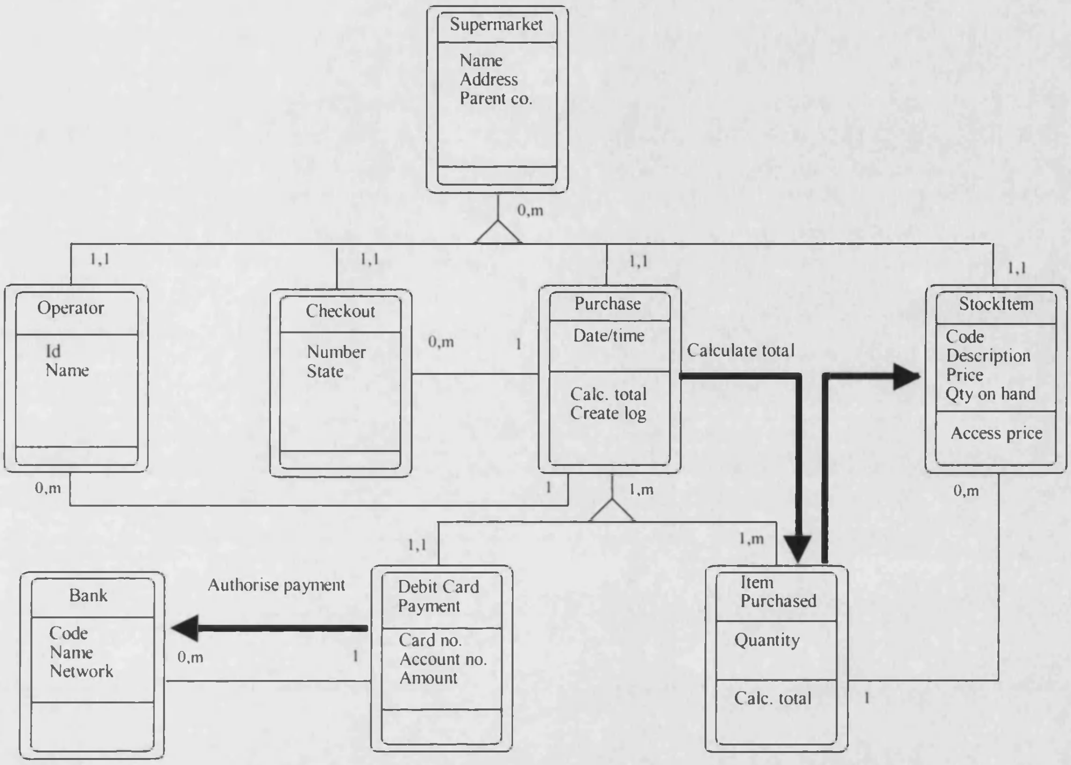


Figure A.8 Object-Oriented Analysis Class-&-Object diagram

Approach to constructing model The OOA model is constructed through a very straightforward application of the steps outlined above. Some informal checking and cross-checking is specified. Many-to-many relationships between classes, instance connections between objects of a single class, and multiple instance connections between the same objects should all be investigated further. OOA leaves the identification of services (operations) until last, concentrating instead on data structures (although, at the discretion of the analyst, services may be identified before attributes).

Assumptions OOA assumes that requirements are readily available and can be described fully. The traditional separation of requirements analysis from design is dispensed with; in OOA, the object/class diagram is intended to be a depiction of the structure of an object-oriented software system, and at the same time, a model of reality with classes representing 'real-world objects'. Hence there is an assumption that reality and software can be modelled identically.

A.8 Problem-oriented approach

The ISAC *(Information Systems Work and Analysis of Changes)* method (Lundeberg 1981) pays close attention to the process of identifying problems and arriving at workable solutions. No assumption is made about any particular solution, and ISAC can therefore be used in situations where no automation is attempted (and even in situations where no information system is required).

ISAC (Lundeberg 1981)

Extent of life cycle addressed ISAC is designed to cover the whole information systems life cycle. There are five stages: (a) change analysis, to examine existing problems and to identify opportunities for new information systems; (b) activity studies, in which the information systems are modelled and alternative implementation strategies are evaluated; (c) information analysis, in which any resulting automated information systems are analysed and specified; (d) data system design; and (e) equipment adaptation.

Steps that apply to conceptual modelling The first three stages are relevant to conceptual modelling. Since ISAC is a 'request-driven' method, it does not begin with a planning phase in the usual sense. Instead, in stage (a) (change analysis) the needs and objectives of different user groups are noted and consolidated. The aim of change analysis is to identify problems in an organisation, to arrive at possible solutions and to identify a preferred solution. The activity in the organisation is modelled and needs are identified by comparing users' objectives with what the existing system provides. Users are encouraged to suggest potential changes to the system which would meet their needs. Their suggestions are analysed and an optimum set of changes is chosen. If part of the proposed change involves the creation or modification of an information system, the method proceeds to the next phase.

In stage (b) (activity studies) the activity models are refined and subsystems are identified which meet the needs of particular user groups. Since (unlike most structured methods) ISAC does not assume that a single information system is the desirable outcome of this process, these sub-systems may well overlap or have conflicting requirements. A cost-benefit analysis is performed on each subsystem, using alternative implementation strategies (called ambition levels) but without thinking of particular technological solutions. The result is a choice of implementation strategy, from which an overall project plan can be drawn up, co-ordinating the development of the different subsystems. Stage (c) (information analysis) is applied only to those subsystems that are intended to be automated. Functions and data are analysed using decomposition techniques, and the analysis is completed by detailed process description and by outlining environmental considerations such as security and data volumes.

Context and representation of model ISAC incorporates several distinct modelling techniques that are not found in other methods. The existing and new systems can be modelled using A-graphs (activity graphs), which decompose in a similar way to data flow diagrams. Using the A-graph, subsystems are identified to meet the needs of particular user groups. The functions of a subsystem are analysed using precedence analysis, a technique in which processes are broken down into a set of simple data transformations, to give rise to an I-graph (information precedence graph). The structure of information being transformed in the I-graphs is modelled using C-graphs (component graphs), a hierarchical decomposition technique which breaks down each information set into its elementary components. Detailed process description is performed using a form of decision table. Figures A.9 and A.10 illustrate A-graphs and I-graphs.

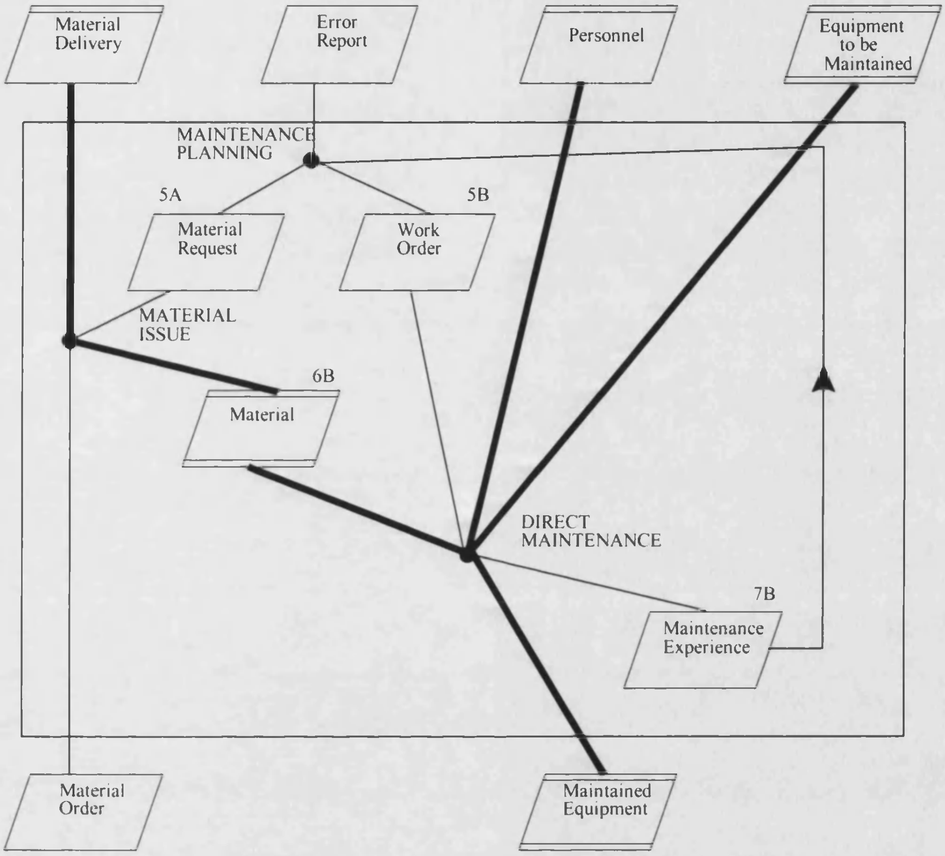


Figure A.9 ISAC A-graph (after Lundeberg et al (1982) p.18)

Approach to constructing model ISAC does not contain the concept of an overall 'corporate' model, since each subsystem is treated on its own merits. Therefore, it is possible to construct the model for each subsystem without the need for cross-checking, and so modelling proceeds in a straightforward fashion as outlined above.

Assumptions The participation of users is considered very important in ISAC. Emphasis is placed on the effects of information systems on their users, to the extent that more usual cost-benefit approaches are considered inappropriate, since they do not necessarily place people ahead of financial gain. ISAC is notable in that it does *not* subscribe to some very common assumptions: it is not assumed that a 'corporate' solution will necessarily result from the design process, and it is not assumed that automation is inevitable. These two differences serve to differentiate ISAC from the bulk of information systems methods.

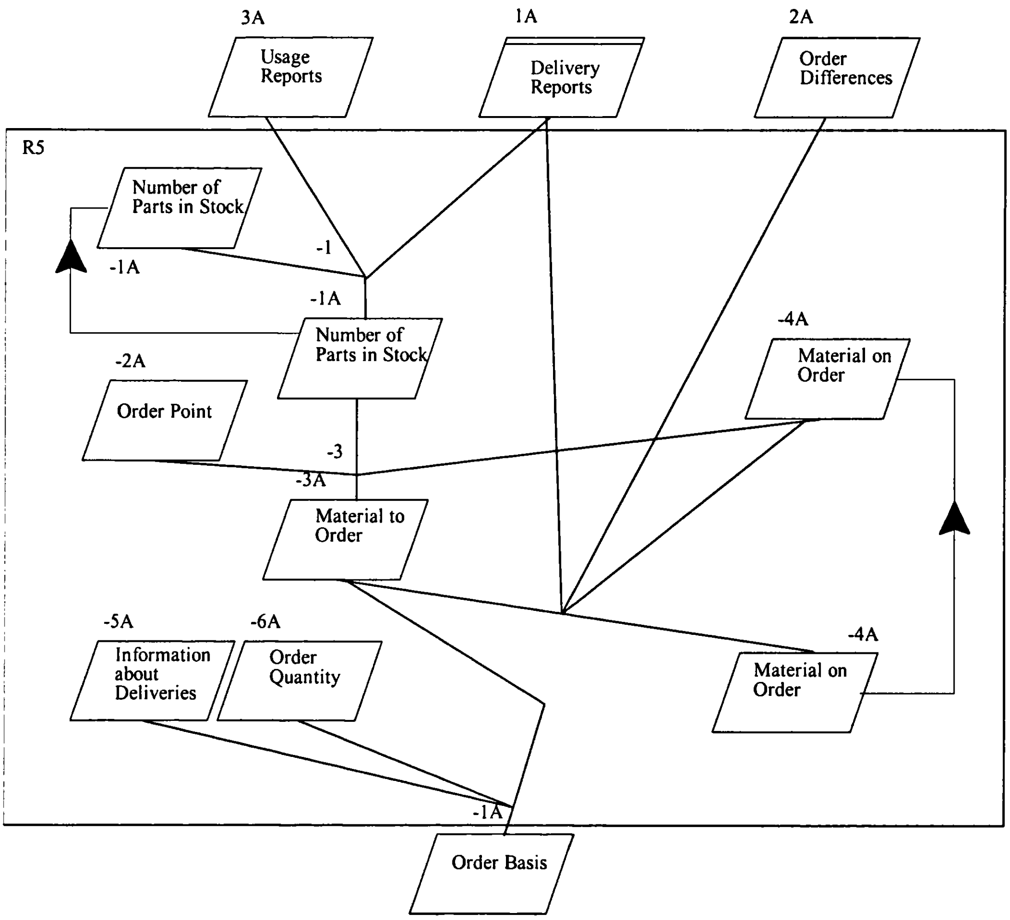


Figure A.10 ISAC I-graph (after Lundeberg et al (1982) p.27)

A.9 Process-oriented methods

The process-oriented approach to information systems analysis involves an examination of business processes or functions carried out in the organisation, and of the data required to perform them. From the statement of business functions a model of required system functionality is derived. Yourdon's *Modern Structured Analysis* was one of a series of 'structured analysis' approaches similar to those used in methods such as SSADM and Gane and Sarson's STRADIS (1979). Central to the structured analysis methods is the idea of top-down functional decomposition.

Modern Structured Analysis (Yourdon 1989)

Extent of life cycle addressed Modern Structured Analysis covers the analysis phase only, although associated techniques for the design phase (known collectively as structured design techniques) are available. The stages are (a) construction of environmental model; (b) construction of behavioural model; (c) construction of user implementation model.

Steps that apply to conceptual modelling All of the stages are relevant to modelling. In stage (a) (construction of environmental model) the overall scope of the required system is defined using a statement of purpose and a 'context diagram' (data flow diagram) that shows individuals, systems and organisations external to the system, and identifies high-level data inputs and outputs from the system. An event list is also constructed, identifying those business events to which the system must be able to respond. In stage (b) (construction of behavioural model) the functions of the required system are decomposed into increasing levels of detail using data flow diagrams. The resulting lower-level processes are cross-referenced to the business events identified earlier, giving rise to partial data flow diagrams showing how the system will respond to each event. Each process is described using pseudocode or some equivalent method of specification. A data model is created using the object modelling technique and a detailed data dictionary is prepared. For systems that exhibit some 'real-time characteristics' state-transition diagrams are prepared showing the relationships between the states of each data entity and business events. In stage (c) (construction of user implementation model) the automation boundary for the new system is decided upon and the user interface is defined.

Context and representation of model Modern structured analysis makes use of several diagrammatic techniques to represent information. Data flow diagrams show the flow of data in and out of processes and to and from data stores (such as files) and external entities (people, organisations or other systems). Figure A.11 illustrates a simple data flow diagram.

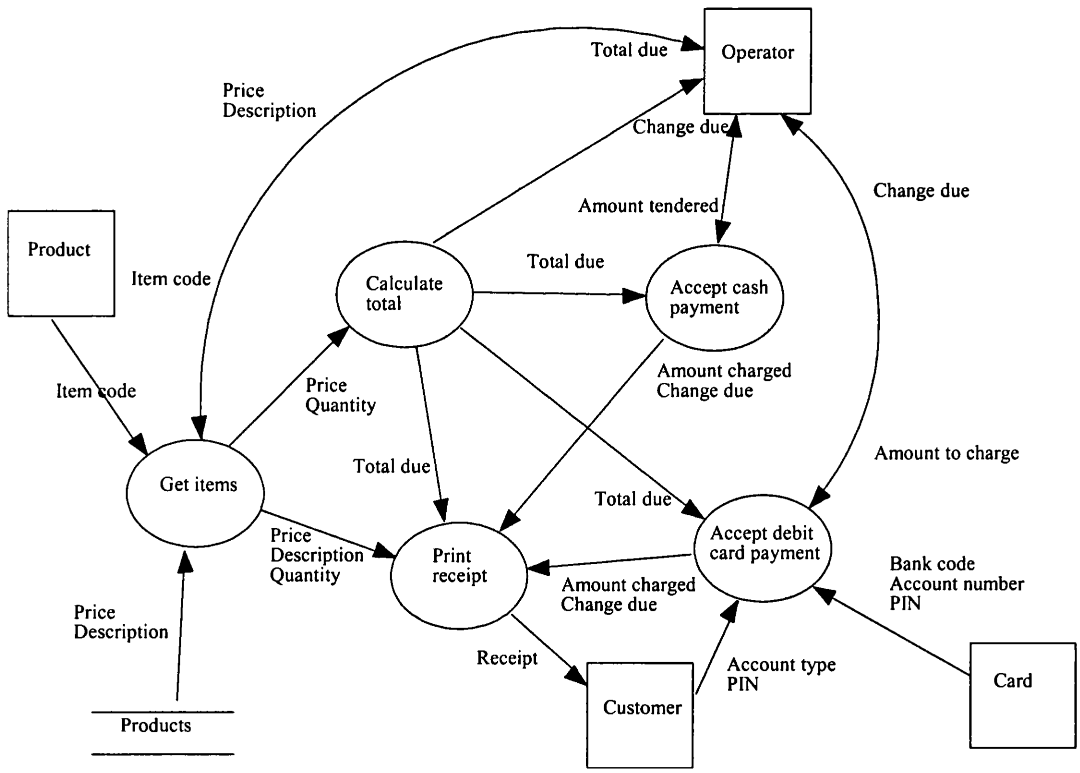


Figure A.11 Structured analysis data flow diagram

This version of structured analysis was notable in that it included high-level depiction of data structures, in the form of entity-relationship diagrams. In appearance an entity-relationship diagram is similar to the logical data structure diagrams used in SSADM, except that relationships between entities are shown using diamond shapes (Chen 1976) rather than simply as lines. It is also possible for relationships to participate in other relationships in the same way as entities. Detailed data structures are expressed in a data dictionary notation using a structured text format.

State transition diagrams are produced for the system as a whole (not for each entity as in SSADM). The state transition diagram shows the states that the system can take and the allowable transitions between states that can result when external events occur. Transitions are defined in terms of a condition (e.g. 'card payment selected') and an action (e.g. 'request amount to charge'). Figure A.12 illustrates a simple state transition diagram. Overall, the environmental and behavioural models form the *essential* model, a statement of what the system must do to satisfy the users' requirements. The essential model is intended to state only *what* the system must do, and to omit any reference to *how* the system will be implemented.

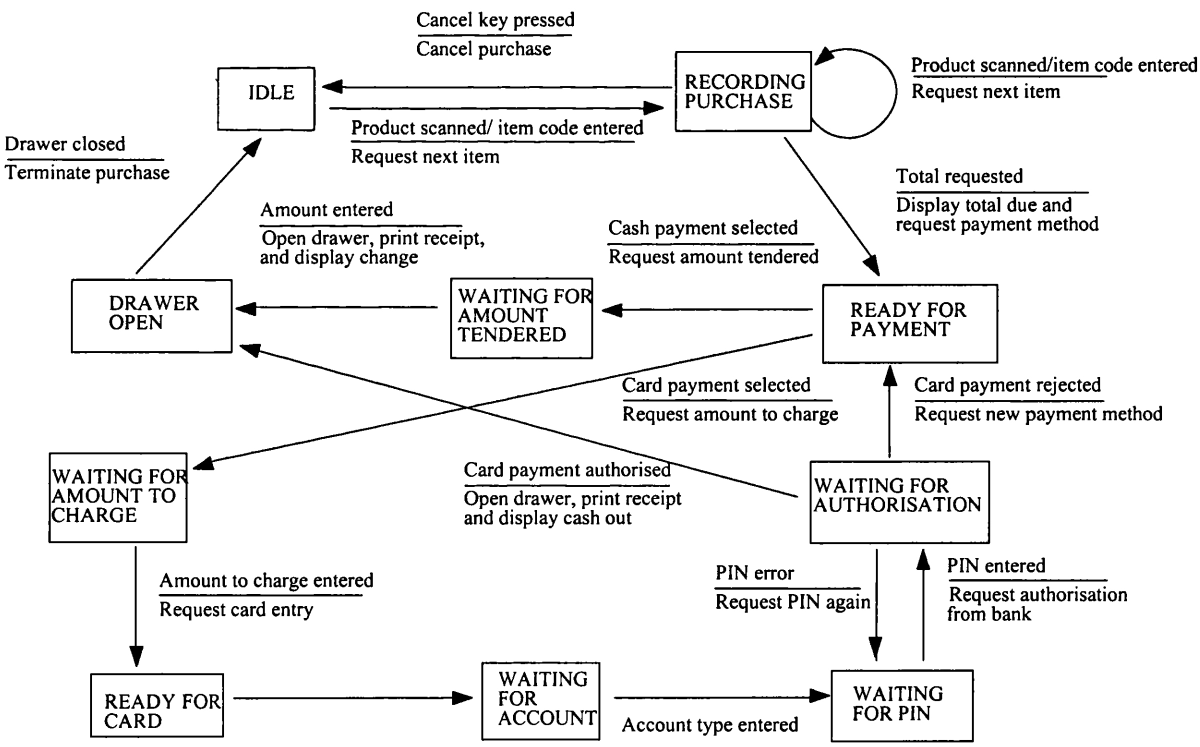


Figure A.12 Structured analysis state-transition diagram

Approach to constructing model Modern Structured Analysis stresses the content of models rather than the particular processes used to arrive at them, and so the life cycle is reasonably flexible. The analyst is urged to consult fully with users and a good deal of advice is given on topics such as interviewing, management and the use of CASE tools. It is acknowledged that cross-checking and correction between the different model components is required before the process is complete.

Assumptions Like many 'structured' approaches, Modern Structured Analysis assumes implicitly that processes and their expected information requirements can be defined in advance with accuracy. In contrast with earlier versions of structured analysis, which concentrated on the process perspective, it is also assumed that analysis from several perspectives (data, processes, and events) is necessary to obtain a compete picture.

A.10 Prototyping

A frequently-repeated complaint about many information systems methods is that they require the user to express requirements in rather abstract terms. The result is that requirements may be incomplete or users may fail to appreciate the full implications of a requirements specification. Either way, the resulting systems may fail to meet the users' real needs. *Prototyping* is aimed at involving users fully in the development process so that their requirements can be made concrete at an early stage. Prototypes are live, working systems, although they may actually simulate part of the processing which they appear to perform. A prototype is refined to arrive at a preferred design. The key advantage of prototyping as a design tool is that a prototype is real; there is no need for a user to imagine what the information system will look like and how it will respond, since the prototype can demonstrate the appearance and behaviour of the system. Through prototyping the implications of particular decisions can be demonstrated, minimising the risk of producing an inappropriate solution.

Rapid prototyping (Maude and Willis 1991)

Extent of life cycle addressed A prototype can perform several functions. It can (a) help to formulate requirements, at a detailed level; (b) test assumptions made during requirements specification; and (c) show how well different design alternatives work. Therefore prototyping is relevant to the requirements analysis, design and construction phases.

Steps that apply to conceptual modelling Prototyping is generally used to elicit and test requirements in two main ways. (a) In *rapid* prototyping, the prototype is discarded once its use in the definition of requirements is complete. Features demonstrated in the prototype may be included in a new information system. (b) In *evolutionary* prototyping, the prototype is enhanced until it becomes a finished product, provided that the programming language being used allows the prototyped application to be sufficiently efficient and maintainable. Prototyping must be preceded by at least a high-level definition of requirements, since it focuses on individual programs rather than complete systems.

Context and representation of model A prototype is itself a model of a portion of the required system, and is useful primarily to demonstrate the appearance and operation of possible user interfaces. Prototyping is less frequently used to gather information about 'hidden' aspects of a system such as data structures or detailed processing logic.

Approach to constructing model Whichever prototyping approach is chosen, the process will be an iterative one. The initial prototype will be extended and refined to produce a succession of new versions, and a prototype may even be completely discarded at some stage in favour of a new one. The aim is to test ideas and alternatives quickly and cheaply, and therefore suitable tools (such as fourth-generation languages) are essential.

Assumptions The key assumption behind prototyping is that it is often impossible for people to adequately formulate their information requirements at the first attempt (Boar 1984). Problem-solving tends to be an iterative activity—an optimal solution to any complex problem is rarely obtained at the first attempt. Instead, partial solutions are refined gradually until they reach a satisfactory state (Avery and Baker 1990). As Berrisford and Wetherbe (1979) observed: *"Most experienced systems analysts agree that it is difficult, if not unrealistic, to ask managers to define their information requirements on paper. Managers must work with a system to appreciate its strengths and weaknesses."* Prototyping is an attempt to correct this deficiency in the way we view the process of creation of information systems.

A.11 Socio-technical approach/participative design

Socio-technical approaches stress the need for attention to be paid during the systems development process to the needs of individuals and to the jobs they perform. This view is supported by the assertion (Bostrom and Heinen 1977) that one of the main causes of information systems failure is an inadequate view of the organisation, which leads in turn to designers ignoring organisational behaviour issues. For instance, systems designers' own notion of responsibility may not include users taking responsibility for their own process of change. Designers may not realise that, in introducing new systems, they are creating secondary changes in the way people work and which may have profound effects on jobs. New information systems are often designed for more efficient task accomplishment, without reference to other effects. Systems whose design is imposed by management may lead to jobs becoming less interesting, less fulfilling or less satisfying.

Socio-technical approaches stress the importance of the social interaction and learning that occurs during information system development. They balance technical development concerns by employing organisational development techniques such as job design, the use of autonomous group structures, team building, and surveys to obtain feedback. The importance of an initial strategic design process is not minimised, but user involvement and empowerment are considered paramount.

The ETHICS (*Effective Technical and Human Implementation of Computer Systems)* method is based on the view that the users of information systems are of great importance in the design of those systems. ETHICS is a participative approach, which means that each of its phases requires extensive user commitment and participation. Users are delegated to steering committees and design teams, and it is these groups which carry out most of the work. The role of the analyst is restricted mainly to assisting users and providing them with the support they need as they carry out the ETHICS process.

ETHICS (Mumford 1996)

Extent of life cycle addressed ETHICS covers the whole information systems life cycle, although it pays particular attention to the earlier stages: requirements, analysis and design. The key tasks (grouped for convenience) are: (a) planning (identifying the need for change, identification of system boundaries, description of existing system, and definition of key objectives and tasks, b) analysis (diagnosis of efficiency needs, diagnosis of job satisfaction needs, future analysis, and specifying and weighting efficiency and job satisfaction needs and objectives, c) design/implementation (parallel organisational and technical design of the new system, selection of technical options, detailed work design, implementation and evaluation).

Steps that apply to conceptual modelling Because ETHICS pays particular attention to the needs of individuals and groups throughout the whole systems development process, each of its stages has some relevance to conceptual modelling. In stage (a) (planning) problems and opportunities are identified to see if there is a need for change. Given the need for change, the design group identifies the boundaries of the system to be designed. A detailed analysis of the existing system is carried out, helping to educate the team in aspects of the system with which they might not be familiar, and this is used to derive a statement of specific problems and solutions. By examining the different areas in the system, and comparing their goals with their current activities, a set of objectives for the new system is produced.

In stage (b) (analysis) particular problem areas in the existing system are described and existing levels of job satisfaction are determined by use of a standard questionnaire. Recommendations for the new system are then made. Potential changes in the new system are envisaged and ways of minimising their impact are outlined. A list of objectives is prepared, ranking the various factors discovered during the ETHICS process in terms of their relative importance. The most important objectives are split into priority objectives (essential) and secondary objectives (desirable).

In stage (c) (design/implementation) changes in the organisation resulting from the introduction of the new system are specified. Principles of socio-technical design are used to ensure that any new or changed jobs are fulfilling for their owners, with attention paid particularly to task variety and job enrichment. These principles can be applied to self managing groups as well as individuals. Technical solutions are also designed, each being subjected to the same criteria as the organisational design options. The organisational and technical design options are merged, and a combination which meets the objectives best is selected. Detailed work design is then followed by implementation. An evaluation process ensures that feedback into the continuing socio-technical design process is achieved.

Assumptions The socio-technical approaches are built on an assumption that many information systems fail because their designers concentrate on technical issues at the expense of all-important social and political ones.

Joint Applications Design (JAD) (Wood and Silver 1995)

Many systems development methods fail to address the need for political and social management of the systems development process. Formal and technical issues are dealt with but the method user is left to rely on his or her own resources to deal with the gaps left by the method. In contrast *Joint Applications Design* (JAD) (Wood and Silver 1995) is a powerful technique for improving the quality and speed of information systems development which *"provides a mechanism for managing the politics of a project, increasing user commitment and involvement through consensus-building and objective and unbiased leadership"* (Andrews 1991).

Extent of life cycle addressed In general, JAD is used for definition of requirements and high-level design. JAD is not an information systems method in itself, but provides a social context and framework within which a method can be applied. JAD prescribes the manner in which sessions are conducted, but leaves the content of the sessions up to the participants.

Steps that apply to conceptual modelling The application of JAD is based around 'JAD sessions' in which the major participants in the development of a new information system—users, management and technical specialists—work together to formulate requirements and to design the system as rapidly as possible. A JAD facilitator guides the progress of the sessions. Techniques such as object modelling, data modelling, prototyping and CASE are often applied.

Context and representation of model Because JAD can be applied with a range of methods, no particular modelling techniques are specified. However, the structure of JAD sessions is centred around the use of graphical and textual models to aid in analysis and help ensure good communication. Flip charts, projectors and white boards are commonly used.

Approach to constructing model JAD is essentially participative, and therefore models are constructed jointly, with the facilitator acting to ensure effective collaboration between the participants. The aim is to achieve good communication and fruitful discussion, with swift resolution of issues.

Assumptions JAD springs from the premise that success in requirements definition and system design depends on more than technical system-related skills. According to Andrews (1991), the JAD facilitator must possess knowledge of JAD process and structure, methods, CASE concepts and diagramming, group dynamics and behavioural psychology, and be skilled in basic facilitation and selling. The facilitation process is concerned at least as much with people and how they interact as with technical system issues. According to Andrews (1991) *"Systems people do not know how to work with people. They know little about the sciences of group and individual behaviour, the structure of effective communications or the skills for promoting involvement and assisting business people in constructing decisions and building consensus."* Although it is widely acknowledged that social issues such as group dynamics and politics play an important role in information systems, few methods actively address these issues. In contrast, *"The power of the JAD technique is found in the integration of behavioural and group dynamics techniques within the structure of a soundly engineered methodology"* (Andrews 1991).

A.12 Strategic information planning

Strategic information planning (or strategic information systems planning) approaches aim to derive information and information systems needs from an examination of the organisation's overall goals and objectives. Battaglia (1991) identifies seven steps to strategic information planning, listed in Table A.5. IBM's *Business Systems Planning* (Martin 1982) is a well-documented and comprehensive approach to strategic information systems planning which has been used in a large number of organisations. In BSP, information requirements are mapped onto the organisation's structure and business processes; no independent information analysis is carried out.

Inventory existing information.
Identify and classify the information needed by the company.
Analyse this information and corporate needs from a strategic level.
Develop a plan for information use that is in line with overall corporate goals and objectives.
Design solutions to effectively meet future information needs
Set standards that allow for the planned creation, production, storage and distribution of information.
Create a budget for creating, storing and distributing information.

Table A.5 Battaglia's strategic information planning steps

BSP (Martin 1982)

Extent of life cycle addressed As a planning method, BSP covers only the very earliest stages of the information systems life cycle, although some of its products (e.g. process-data matrix) are useful at later stages. The stages of BSP (grouped for convenience) are: (a) preparation (gaining commitment, preparing for study, starting the study, b) gathering information (defining business processes, defining business data, defining an information architecture, analysing current systems support, interviewing executives, c) analysis (defining findings and conclusions, determining architecture priorities, reviewing information resource management, d) conclusions (developing recommendations and an action plan, reporting results, overview of follow-up activities).

Steps that apply to conceptual modelling All of the steps of BSP are relevant to conceptual modelling. In stages (a) and (b) (preparation and gathering information) the organisation is modelled from several points of view so as to gain an accurate picture of current and future activities and needs. In stages (c) and (d) (analysis and conclusions) plans are drawn up to allow the needs to be met through the provision of suitable information systems.

Context and representation of model In BSP matrixes are used to show important relationships between model components. An organisation-process matrix is constructed, identifying who in the organisation is involved in each process. It is generally expected that between twenty and sixty processes will be obtained for the whole business, and these are grouped into between four and twelve sets. The data entities used, created and controlled by the processes are identified using a process-data matrix. The matrix is reorganised to show clusters of processes which use related data. Each cluster represents a sub-system in the organisation's overall information system. The set of subsystems and their interconnections represents the organisation's information architecture. A process-data matrix is shown in Table A.6. In the matrix, one or more characters is placed in each cell, indicating the ways in which the process uses the particular class of data. The characters are: C (indicates the function creates data of this type), R (the function reads data of this type), U (the function updates data of this type), and D (the function deletes data of this type). The diagram is sometimes called a 'CRUD matrix'.

Approach to constructing model The organisation's activities are first analysed in order to create a stable list of fundamental business processes, assembled by consulting management and from other sources. The products or services of each business unit are defined, and the resources (money, personnel, materials and facilities) required to produce the products are identified. These are used to check the process list, considering a product life cycle with four stages: (a) requirements, planning, measurement and control, (b) acquisition or implementation, (c) stewardship, (d) retirement or disposition. The resulting processes are classified as strategic planning, management control and operational control activities. The processes are grouped and recombined, and are then represented on an organisation-process matrix and clustered process-data matrix as described above. The clustered process-data matrix is used to develop the detailed information requirements for the organisation as a whole.

Data subjects	**Finance, accounting & control**	**Human resources & training**	**Production & production planning**	**R&D**	**Marketing & product planning**	**Order processing**	**Business management**
Employee/HR plans and reqs	R	CRUD	R		R		R
Financial plans	CRUD	R	R		R		R
Income, outflow, investment	CRUD				R		R
Product plans	R	R	R	R	CRUD		R
Product materials	R		RU	CRUD	R		
Finished products	R		RU	CRUD	R	RU	R
Company locations/facilities	R	R	R	R	R		CRUD
Equipment	R		RU	CRUD	R		R
Organisational plans	R	CRUD			R		CRUD
Vendors	R		CRUD			R	
Customers	R				R	CRUD	R

Table A.6 BSP process/data matrix

Assumptions BSP and other strategic planning approaches are based on the assumption that the information (and information systems) required by an organisation can be ascertained through a high-level examination of business goals and functions; that is, in a purely 'top-down' manner. The needs of particular individuals are not considered important enough to be taken into account in this process. However, it should be stressed that BSP is primarily a method for planning, and that its use typically is followed by more detailed investigation of requirements. At this stage the needs of particular individuals may well be taken into account.

A.13 Systems approach

Methods based on systems thinking take ideas from general systems theory. The reductionist approach commonly applied to systems analysis is rejected in favour of a more holistic (or 'systems') view. The complexity and 'fuzziness' of organisational systems are considered too great to be dealt with effectively by structured techniques such as decomposition. The *Soft Systems Methodology* (SSM) is not designed specifically for use with information systems, but is a generalised organisational problem-solving approach. As such, it makes no assumption that the solution to particular problems will contain information systems. SSM takes as a starting point a 'problem situation' which is effectively a subset of an existing system that is perceived to contain problems. The aim of the method is to explore the problem situation, to identify underlying problems (which may well not have been perceived initially) and to suggest alternative ways of looking at these problems which could be used in solving them. The approach taken in SSM is therefore quite distinct from that taken in strategic planning-based methods, where a more simplistic view is taken of the organisation, its objectives, and how to achieve them.

SSM (Checkland and Scholes 1990)

Extent of lifecycle addressed SSM is not specifically an information systems method, and therefore its applicability to the information systems life cycle is not straightforward. The seven-stage approach is as follows: (a) the problem situation: unstructured; (b) the problem situation: expressed; (c) root definitions of relevant systems; (d) conceptual models; (e) comparison of conceptual model with problem situation; (f) feasible, desirable changes; (g) action to improve the problem situation.

Steps that apply to conceptual modelling The method can be applied as a form of planning tool, identifying areas of the organisation in which change is desirable and feasible. In this sense SSM can be seen as being directed primarily at the early stages of the information systems life cycle (this is how SSM is used in Multiview). Alternatively, SSM could be applied to a particular information system, and in this case is relevant to the whole life cycle, although it includes no component specifically directed at the technical aspects of creating computerised information systems. In stage (a) (the problem situation: unstructured) information about the problem situation is obtained to help identify the scope of the exercise. This information is structured in stage (b) (the problem situation: expressed). The analyst draws a 'rich picture'—a semi-formal representation of the problem situation, showing elements such as people, groups, and problems (Figure A.14). The rich picture is intended both as a vehicle for communication between analyst and system users and as a means of stimulating discussion about important issues. It is hoped that use of the rich picture will help to generate awareness of the main problem themes.

In stage (c) (root definitions of relevant systems) ways of looking at the situation which may be helpful are identified. The literature on SSM avoids the use of the term 'solution' in this context, but it is clear that the definition of relevant systems must be driven by some insight into possible ways of solving the problems which have been revealed in the previous steps. One particular relevant system is then chosen for further consideration and a root definition of this system is established. The root definition is checked using the 'CATWOE' checklist (customers, actors, transformation, weltanschauung (worldview), owner, and environmental constraints).

In stage (d) (conceptual models) a 'conceptual model' (Figure A.13) of the chosen relevant system is constructed using a semi-formal notation. The conceptual model is not intended to be a model of the real world, but instead represents a conceptualised process model showing how the chosen relevant system might work, given the root definition established for it. In stage (e) (comparison of conceptual model with problem situation) the conceptual model of the relevant system is compared with the model of the existing system prepared earlier. From this comparison is drawn a set of recommendations for change. The feasibility and desirability of the recommended changes are analysed in stage (f) (feasible, desirable changes), and in stage (g) (action to improve the problem situation) actions are planned, based on the recommended changes.

Context and representation of model SSM provides a framework within which the analyst has considerable freedom to choose the most appropriate information to capture and the most effective representation to employ. Checkland and Scholes (1990) give several techniques (mainly textual) that can be used. Two graphical techniques are offered: conceptual models and rich pictures. The conceptual model illustrates, using a semi-formal 'box-and-arrow' notation, how a system might work, as shown in Figure A.13. The conceptual model is effectively a task breakdown for the system in which precedence relationships or information flows are shown.

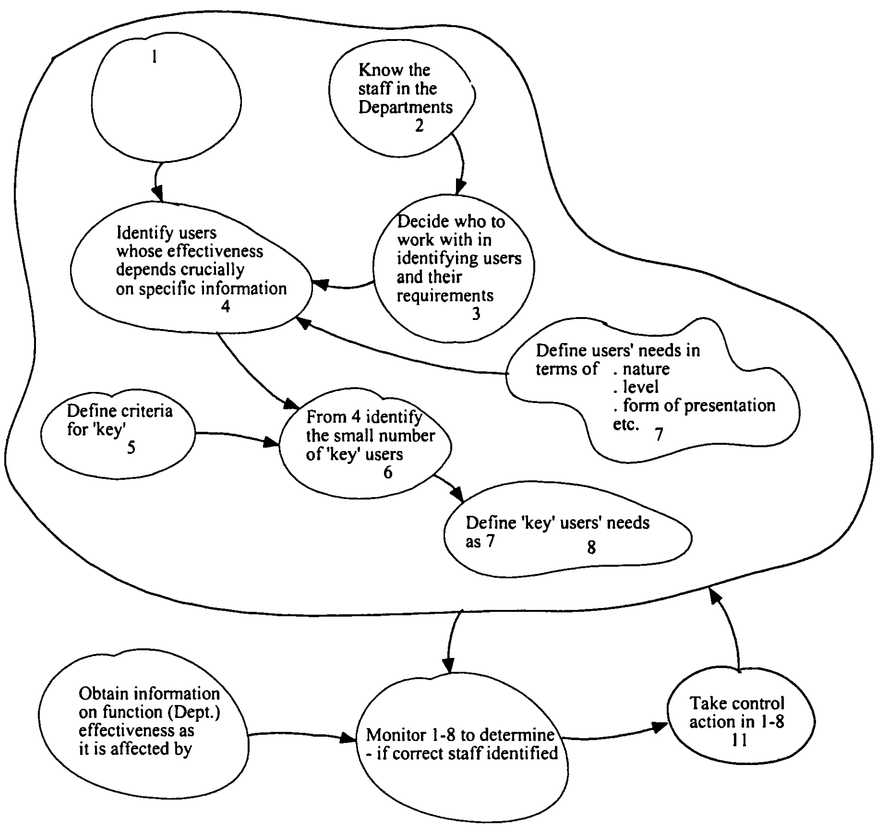


Figure A.13 SSM conceptual model (after Checkland and Scholes (1990) p.71)

Rich pictures use a mixture of iconic symbols and text to illustrate a problem situation. Again, the notation is not formally defined, and it is up to the analyst to depict the situation in whatever way seems appropriate. Rich pictures typically show the main participants in the system (represented as stick people), conflicts (crossed arrows), issues (thought bubbles), scrutiny (eyeballs), and real things, structures or processes (represented as themselves). Figure A.14 gives an example of a rich picture.

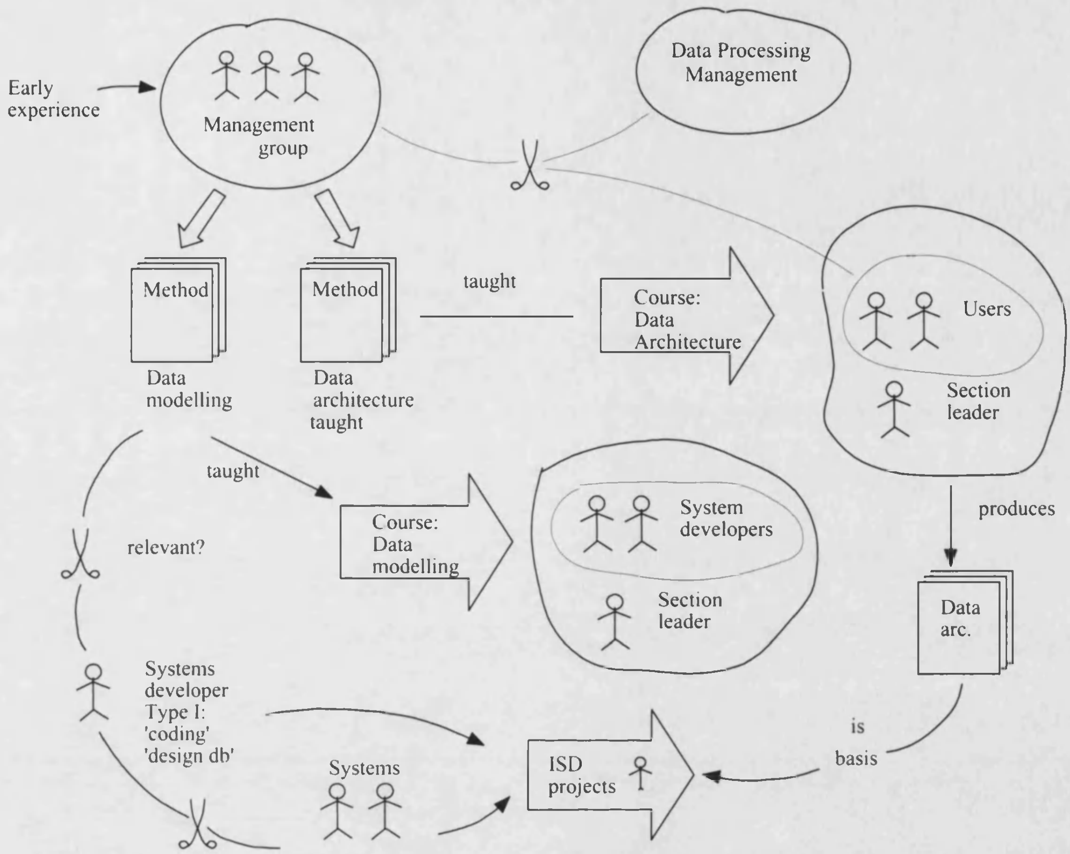


Figure A.14 SSM rich picture (after Checkland and Scholes (1990) p.47)

Approach to constructing model The literature on applying SSM is wide and varied, with many possible approaches set out. In general, SSM is carried out in a participative way, with full involvement of the problem owners, customers and so on. However, a very wide degree of freedom is available to the analyst to choose the most appropriate approach.

Assumptions Fundamentally, SSM is a problem-solving approach, and therefore assumes that some examination of the 'problem situation' will be worthwhile. SSM is also predicated on a view that the 'divide and conquer' strategy used in many methods, where a problem situation is decomposed into smaller problem situations that can be solved individually, is less useful than a holistic approach which considers the overall effect of the system. Having said that, the 'conceptual models' produced in SSM are effectively equivalent to process breakdowns and so do, in fact, use decomposition.

Wilson's *Information Systems Analysis Methodology* (discussed below) is one way of applying SSM specifically to information systems.

Information Systems Analysis Methodology (Wilson 1990)

Extent of life cycle addressed Wilson's approach involves an analysis of organisational information requirements, and therefore covers the requirements analysis part of the life cycle. The approach is divided into seven stages: (a) Primary task description; (b) primary task (conceptual) model; (c) analysis of required information inputs and outputs; (d) construction of upper half of 'Maltese cross' (Figure A.15) showing required information; (e) completion of Maltese cross showing existing information; (f) identification of required information processing procedures; (g) activity models for required information processing procedures; (h) map information requirements onto organisation.

Steps that apply to conceptual modelling All of the steps are relevant to conceptual modelling. In stage (a) (primary task description) a root definition is prepared that focuses specifically on the main or 'official' purpose of the organisation. It is hoped that choosing the most straightforward root definition will lead to a model that bears a close resemblance to reality. A corresponding conceptual model is developed in stage (b) (primary task model) and validated by comparison with the organisation itself. In stage (c) (analysis of required information inputs and outputs) each activity in the primary task model is analysed in terms of its information inputs and outputs. A hierarchical data decomposition is prepared showing the information structure (similar to that used in ISAC).

Stages (d) and (e) (Maltese Cross) involve the production of a matrix showing information inputs and outputs. The upper half of the 'Maltese cross' is first prepared, showing which pieces of information are input or output to which activities from the primary task model. The Maltese cross is completed by showing, on its lower half, the inputs and outputs of each existing information processing procedure. Comparison of the top and bottom sections of the Maltese cross can reveal shortcomings in the existing information system. Possible ways of resolving these shortcomings are investigated.

In stage (f) (identification of required information processing procedures) new or modified information processing procedures resulting from stage (e) are described in detail. Corresponding activity models are prepared in stage (g) (activity models for required information processing procedures). Finally, in stage (h) (map information requirements onto organisation) the processed data provided by the new system is related to the needs of the users, based on a definition of their roles in the organisation.

Context and representation of model The primary task description and primary task model are prepared largely as described for the root definition and conceptual model in SSM. Wilson's main contribution in this method, over and above the features and capabilities of SSM, is in the use of the Maltese Cross, which relates information inputs and outputs to activities in the organisation (in the top half) and to existing information systems (in the bottom half). The Maltese Cross can be used to identify shortcomings in existing information provision by comparing top and bottom halves. Figure A.15 illustrates a Maltese Cross.

	x		x					x				Decide to accept or reject												x
		x		x	x		x	x								x							x	
						x	x	x				Test the sample				x						x		
									x	x	x	Take sample of product				x								
Decisions	Comparison results	test results	Decision criteria	Standards	Specifications	Test results format	Test routine	Product/sample id	Location information	Sampling routine	Product id		Product id	Sampling routine	Location information	Product/sample id	Test routine	test results format	Specifications	Standards	Decision criteria	Test results	Comparison results	Decisions
					x						x	PRODSPEC	x						x					
							x				x	PRODTEST	x				x							
										x	x	SAMPROD	x											

Figure A.15 Maltese Cross (after Wilson (1990) p.239)

Approach to construction of model The approach recommended by Wilson is essentially the same as for SSM proper, and selected users and other system participants are involved fully in the production and refinement of models. The method steps are followed in a linear sequence.

Assumptions Although based on SSM, Wilson's approach is significantly closer in form to traditional systems analysis methods, making the fundamental assumption that an analysis of information requirements is the first priority. However, Wilson indicates that the first stage (preparing a task description) should be preceded by an issue-based analysis such as is carried out when SSM is used. It is also assumed that specific information requirements can be identified for given activities.

A.14 Conclusions

The information contained in this appendix is used throughout the rest of the thesis as a reference and as a basis for comparison. The methods described here vary in many ways. For practical reasons the experimental part of this research focuses only on a narrow range of methods, which are similar to the object-oriented (Section A.7) and data-oriented (Section A.4) approaches described above. But the psychological principles in Chapter 3 are formulated to have quite general application and may be found useful in a wide range of methods.

Perhaps the area of greatest diversity between the methods discussed here is in deciding what information is most important or relevant. In SSADM, analysis focuses on an examination of data structures, processes, entity life histories and so on, all of which have obvious links to the data bases and computer programs that are likely to be contained in the resulting computer software. Like many widely-used 'structured' methods, SSADM creates an abstract (or 'logical') model of the organisation before implementation considerations are brought into play, but nevertheless concentrates mainly on requirements for the target computerised information system. The object-oriented approach takes a similar view, but dispenses with the distinction between conceptual and implementation models, representing the organisation directly in terms of software objects which communicate solely by passing messages. Information Engineering also takes a broadly similar view, but stresses the need for a strategic, 'corporate' analysis of objectives and resulting information needs before any detailed analysis is performed.

Other methods take quite different views. For instance, in ETHICS far more attention is paid to the effects of any new system on individuals' jobs and the workings of groups. In prototyping-based methods, very close attention is paid to the detailed workings and user interfaces of particular programs, at the expense of a more 'global' view. In Multiview, an attempt is made to cover all perspectives by incorporating elements from several quite distinct methods.

Many different types of model are used in the methods summarised in this chapter. They range from simple textual lists (such as the requirements catalogue used in SSADM) to complex structures involving text and graphics (such as ISAC's A, I and D-graphs). They can be relatively informal (e.g. SSM's rich pictures) or relatively formal (e.g. Rumbaugh's state diagrams). They can be used early in the life cycle (e.g. Information Engineering's enterprise model) or later (e.g. user interface prototyping). What these techniques have in common is that they allow *relevant* aspects of the organisation, or its requirements, to be modelled, whether it be information structures, processes, issues, problems, ad hoc needs, or jobs. Modelling relevant aspects of the organisation serves several purposes: (i) it helps the analyst gain a good understanding of the organisation; (ii) it aids communication between participants by focusing attention and providing a succinct way of representing information; (iii) it directs attention to particular aspects of the problem situation. Modelling an organisation can help identify areas in which change is feasible or desirable, and the models produced during requirements analysis can be used in later stages, saving duplicated effort.

Almost every method that seeks to capture models of any complexity does so by representing them graphically. There is good reason to do this; as the saying goes "a picture is worth a thousand words". A graphical representation can convey complex relationships at a glance. The graphical representations used by the methods examined in this chapter vary from tabular representations (e.g. matrices in BSP) to more intuitive pictorial depictions (e.g. SSM's rich pictures). Perhaps the most popular type of graphical notation is the 'box-and-arrow' style used in data flow diagrams, entity life histories, object model diagrams and so on. These notations are graphical rather than pictorial, and they use rather arbitrary symbols to convey meaning. In most cases, the choice of symbol and notation provides no clue to the reader as to the meaning of the diagram (there are some exceptions, such as the use of the 'crowsfoot' in SSADM logical data structure diagrams to indicate "many"). The reader must therefore learn the graphical language before being able to use it. This sort of notation is essentially logical in nature and arguably appeals most to those with logical or analytical frame of mind.

Most of the methods summarised in this chapter also use graphical models as a way of *communicating* information. For instance, in SSADM, the analyst prepares a logical data structure diagram and may well show the diagram to other people to determine its correctness. In Soft Systems Methodology, the rich picture serves as the basis for a dialogue between analyst and other participants. In JAD, the concept of visual communication as the basis for dialogue and collaboration is exploited fully, to the extent that JAD sessions make extensive use of projectors, white boards with magnetic symbols, flip charts and so on. Even the seating layout in a JAD session is designed to maximise interaction using visual media. Many methods assume that production of models is primarily the responsibility of the analyst, although most urge the analyst to consult fully with users and other interested participants. Of the methods examined in this chapter, only ETHICS places responsibility for production of models firmly in the hands of users.

Early views of the systems development process were based on the engineering 'project' approach, in which one stage is followed by the next in a sequential fashion. Construction of a new system was preceded by a design phase, which itself followed a requirements analysis phase. The results of each phase were deemed to be largely definitive and any subsequent changes to early results would be negligible. This view is characterised by the 'waterfall' life cycle (Gane and Sarson 1979). This view rests on the assumption that requirements can be expressed fully and correctly before any design or construction is undertaken. For the simplest and smallest information systems this may well be the case, but for the vast majority of information systems, in the majority of organisations, experience has shown that it is not (Vonk 1990, Bubenko 1986). As a consequence several alternative approaches have developed. The increasing use of prototyping is one example. The 'spiral' model of information systems development (Wirfs Brock 1990) is an attempt to formalise the idea that requirements are refined through experimentation. Many methods are based on the earlier 'waterfall' model and therefore make little allowance for subsequent refinement of models, whether at requirements, analysis or design stages. Revision of models is often difficult or impossible simply because of the bureaucratic overhead created by the method. Supporting software tools can help to alleviate this problem by making changes easier to carry out and their potential impact easier to access.

Appendix B
Relevant Psychological Theory

B.1 Introduction

Psychology is *"the study of the mind, as deduced from behaviour"* (Hawkins 1988). This appendix summarises the results from a review of the psychological literature. The aim is to outline findings and theories that can shed light on the practice of conceptual modelling and can be used to formulate psychological principles for use in the experimental part of this research (see Chapter 3). Psychology is relevant in this study because it can help us to understand how models are perceived and how external representations of conceptual models, such as diagrams, relate to the internal mental models that we construct of business situations and of conceptual models. Psychology can provide useful insights into how and why people work the way they do when creating and using conceptual models. It also has something to say about the nature of group interaction in the conceptual modelling process, which often involves collaboration between individuals. Table B.1 and Figure B.1 illustrate these areas of psychology and their relevance to conceptual modelling.

Perception	Models must be perceived in order to be interpreted. Modelling techniques can be designed to exploit perceptual mechanisms.
Comprehension	An understanding of the cognitive processes involved in comprehension can help us make conceptual models more understandable.
Mental models	Conceptual modelling can be seen as an attempt to capture the essence of users' mental models of a business area. Hence we must appreciate the nature of mental models and their relationship to conceptual models.
Memory	An understanding of the mechanisms behind memory may allow us to help reduce the load placed by modelling on memory.
Group psychology	Conceptual modelling is normally a group effort. Group dynamics and its implications are relevant to optimising group modelling performance.

Table B.1 Relevant psychological areas

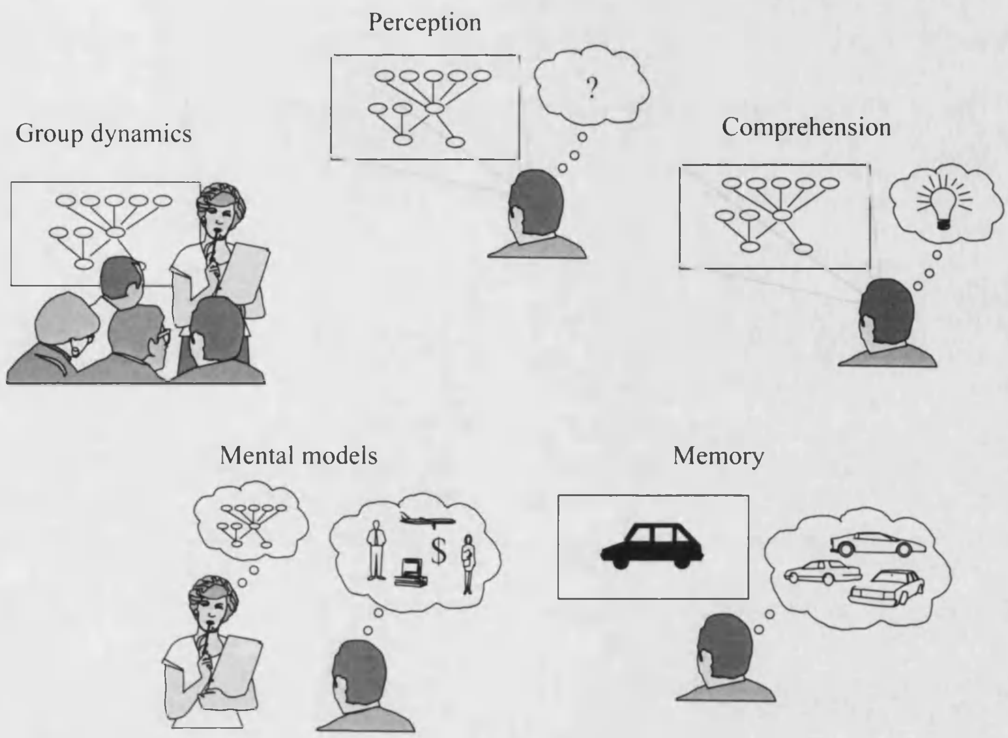


Figure B.1 Relevant psychological topics

Many of the psychological concepts that we shall consider are part of cognitive psychology, the study of *"processes and activities used in perceiving, learning, remembering, thinking, and understanding"* (Ashcraft 1998). This branch of psychology is concerned with *"representation of human knowledge and its use, as seen in human action"* (Malim 1994). Cognitive psychology has developed over the past few decades into a mainstream branch of psychological research. For many years, however, inquiring into the nature of thought was not considered a suitable topic for psychological research. The dominant behaviourist school held that only observable behaviour was worthy of attention and to study inner mental processes was 'unscientific'. Behaviourism explains learning in terms of *operant conditioning*: the modification of behaviour through reinforcement of selected responses to stimuli (Skinner 1950). Operant conditioning offered a plausible explanation for the behaviour of rats in mazes and salivating dogs. As a way of understanding complex human action, the behaviourist approach is arguably less effective, especially with regard to the development of language and mental conceptual structures.

Although behaviourism dominated psychological thought for much of the present century, significant work was done before the rise of behaviourism on what are now termed cognitive processes. Notable amongst this work was that of the *Gestalt* psychologists (Kohler 1947) who pointed to similarities between problem-solving and perception. In their terms, the end results of both perception and thinking were 'gestalts': mental structures that somehow brought together the elements of a problem or a situation into a coherent whole. They emphasised the value of insight in problem-solving, observing that the solution to a problem often comes intuitively some time after initial exposure to it; in other words, after a period of incubation. Perhaps the single aspect of problem-solving considered most important by the Gestaltists was the way in which we represent problems mentally. Experimental evidence confirmed that the choice of external or internal problem representation is an essential factor governing success in problem-solving. Many of these early ideas have a direct relationship to recent research in cognitive psychology that we shall examine later in this appendix (Best 1999).

Like most disciplines, psychology is not a single body of knowledge but includes competing ideas that may contradict one another. Several areas of psychology visited in this appendix contain rival theories that try to explain observed behaviour in different ways. There is a core set of issues on which psychologists tend to concur; for instance, it is generally accepted that short-term memory uses some form of auditory coding (see section B.2). We can safely use results in these areas. But we must be more circumspect in more contentious areas, such as the propositional representation of knowledge in long-term memory (Green 1996).

The remainder of this appendix is divided into four sections. The first covers human memory and its role in the formation of concepts. The second looks at perception and comprehension. The third is concerned with mental models and theories about the ways in which we structure and represent concepts internally. The fourth discusses the properties of group work. These topics are closely inter-linked. For instance, it would be impossible to study comprehension without understanding how things are perceived and how information is held in memory; perception is known to be directly influenced by the way memory works and also by our own mental states, which are themselves affected by working in groups. Hence some topics will inevitably be visited more than once in the course of this appendix.

B.2 Memory

Overview

Most psychologists agree that memory has at least two distinct components: *short-term* and *long-term* memory. Short-term memory is of limited capacity and extremely volatile; information is lost within seconds if not refreshed by conscious repetition (Pinel 1993). Short-term memory is used to store the sensory information that one's attention is currently (or has recently been) focused on. It is thought that this information is held predominantly in verbal or acoustic form (Solso 1998). Long-term memory is rather different in nature. Retrieval generally takes longer (often far longer) than retrieval from short-term memory, and the capacity of long-term memory appears to be almost infinite. Information stored in long-term memory has a 'semantic coding' as opposed to the acoustic coding of short-term memory. What is stored in long-term memory is, in some way, a direct representation of meaning, rather than one that depends on the form in which the information was expressed (Green 1996). Figure B.2 represents the interaction of short-term and long-term memory.

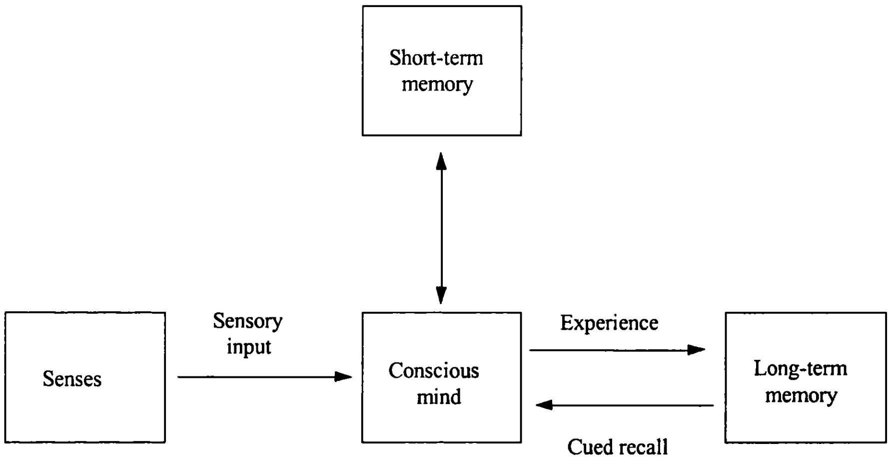


Figure B.2 Simple model of memory

There is no direct physiological evidence for the distinction between short and long term memory, and other models for memory have been proposed. The *working memory* model (Baddeley 1986) presents a more complex view in which short-term memory is made up of several interacting components, with separate processing of acoustic and visual information (Malim 1994). Baddeley's model requires the co-ordination of working memory components by a central executive unit. Another view relates the retention of information to *depth of processing* rather than to any inherent difference in memory mechanisms. According to this view, the more a piece of information is manipulated the more likely it is to be retained. Information processed at a deep or semantic level (that is, information whose meaning is fully considered) will be retained for longer than that processed at a shallow level (Thapar and Greene 1994).

Long-term memory

One of the most remarkable characteristics of long-term memory is its ability to relate ideas together. Long-term memory is connection-rich, or *associative* (Quinlan 1991). Memories are stored not in isolation but with many links to related memories. The nature of these connections can be enormously diverse. Retrieval of information is possible because of these connections, and it is thought that this ability to make and use connections lies at the basis of intelligence (Bechtel and Abrahamsen 1991). *Associative recall* of information is prompted by the use of suitable cues. Unlike the memory of a computer system, where connections must be planned and made explicitly, our brains automatically forge connections between areas of knowledge where some commonality is perceived. For instance, when we learn new facts we inevitably link them to our memories of the physical surroundings in which learning occurs. Research has shown that recall of information is improved if it takes place in the same setting as the original learning (Smith et al 1978). This is an example of cued recall. The fact that we can so easily recall information via its meaning and its relationships to other information tells us that our long-term memory must use a semantic coding. So, in the case of everyday events that have been committed to long-term memory, what is stored is the significance of each event rather than its precise details. There has been some debate about whether or not long term memory uses some form of image representation in addition to semantic coding. But, in general we can say that the information held in memory is stored not in the form of words, images or other symbols, but as some more direct representation of meaning. This explains how, for example, the meaning of two equivalent words can be related even though they may sound and look unlike one another.

Recent computer simulations of brain-like neural networks have hinted at ways in which long-term memory might operate (Khanna 1990). It appears that each fact or item of information is stored not at a single location but instead throughout whole regions of the brain. Clinical evidence supports this conclusion, since severely damaged brains can continue to operate with relatively minor impairment in function. It seems that the billions of connections between brain cells collectively provide memory; no single cell or connection is individually responsible.

The capacity of long-term memory is enormous and long-term memories can last a lifetime. It often seems that the brain is capable of retaining a complete record of an individual's experience, in apparently enormous detail. But evidence suggests that a great deal of the detail supposedly recalled about a situation is added, during retrieval, fleshing out the remembered experience (McCloskey and Egeth 1983). When no specific information is available we tend to recall information of a general nature. This unconscious process is inevitable and involuntary. It provides us with the illusion of total recall even through retention is incomplete. We fill the gaps in our memory with details that are plausible but not necessarily accurate. For this reason eyewitness accounts, such as those given in legal proceedings, are very often unreliable. Although a few gifted individuals are equipped with near-perfect memories, most of us have some difficulty in absorbing and recalling large amounts of information as quickly and as accurately as we might like or believe to be the case.

One interesting characteristic of long-term memory is interference, where a new memory affects or is affected by an existing memory. It can be more difficult to learn a new way of doing something than it was to learn how to do it in the first place. As new memories are laid down, they tend to disrupt older memories if the subject matter of old and new memories are related in some way.

One way of improving our ability to recall information is to organise the information into *categories* (Bourne et al 1979). Considering the connection-rich nature of long-term memory this is hardly surprising, since placing several items in the same category means making a connection between them. Another very effective aid to memory is *imagery*. Associating verbal information with visual information makes it easier to recall (Brandimonte, Hitch and Bishop 1992). Those intending to provide artificial support for memory should beware, however. Research on the use of external aids to memory, such as computer databases, has shown that people are generally reluctant to use external information retrieval mechanisms if they can possibly manage to rely on their own memory. An important factor is whether the user can easily find out what information is stored and, especially, how relevant or useful it might be (Schönpflug 1986). In other words, people may tend to ignore information sources if they anticipate any difficulty at all in obtaining relevant information easily and quickly.

Contents of long-term memory

The semantic coding of long-term memory allows us to construct an internal representation of reality. This representation links knowledge in ways that help us to categorise our experiences, draw inferences, and interact appropriately with our environment. Because of the semantic coding of long-term memory, the process of understanding is intimately connected with the way in which this type of memory operates—a fact that artificial intelligence or cognitive science research has not always taken into account (Green et al 1996, Dreyfus and Dreyfus 1988). Later in this appendix we shall look at possible structures for semantically coded information in long-term memory. But first we shall examine some other types of representation thought to exist in long-term memory.

It is believed that long-term memory has an *episodic memory* component, which stores details of recent or isolated experiences. This memory is analogous to a video recording of an incident, in that it contains a record of everything that has happened, in some detail, but offers no interpretation; little of the meaning or structure of the incident has been abstracted out. Experiences that remain unique and are not repeated may be remembered in great detail thanks to episodic memory. But once an experience has been repeated several times, the record in long-term memory becomes a more generalised 'script' or 'schema', and detail will typically be lost. This is why it is easy to remember where you parked your car the first time you visit a new place, but much harder if you visit the same place frequently. The repeated experience of parking allows the mind to create a generalised script for parking at that location.

The generalisation process is an example of interference. Some see it as a necessary 'housekeeping' operation carried out by the brain so as to avoid unwanted storage of unconnected experiences. The obvious advantage of generalising about experiences is that it makes knowledge more transferable and allows analogies to be drawn (Holyoak and Thagard 1995). Schank (1985) distinguishes between three levels of generality in memory: event memory, situational memory and intentional memory. He gives the following examples of each type:

Type of memory	Example
Event memory	Going to Dr. Smith's dental appointment last Tuesday and getting your tooth pulled.
Situational memory	Going to a health professional's office.
Intentional memory	Getting any problem taken care of by a societal organisation.

Table B.2 Schank's memory types

An ongoing debate surrounds the issue of *images* in long-term memory. There is no doubt that it is possible to recall visual images from experiences that took place in the distant past. For example, it may be easy to visualise a person's face, even if you have not met the person or seen a photograph of them for months or years. Therefore images must be stored in long-term memory in some form. But the images produced in this way are not like the normal visual images produced by looking at objects; some people report that the images they recall have no 'visible' component at all (Kosslyn 1994). It is not clear whether the brain stores images literally (i.e. photographically) or in some other way. Evidence suggests that we do not store photographic images but instead reconstruct images at the time they are required. In other words, the mind's eye sees images that are built up using knowledge of how they ought to (or might plausibly) look. This knowledge is often not specific enough to reproduce an image accurately and so the 'remembered' image may vary considerably from the original. As previously noted, recalled information usually contains plausible embellishments since we tend unconsciously to reconstruct missing information. For this reason, recalled images can be very undependable.

Despite the fact that images seem to be stored in an interpreted rather than literal form, it is possible that they are processed by the brain independently of other types of information. Paivio's *dual coding theory* (1971) postulated the existence of two distinct though interrelated systems for verbal and non-verbal information. Non-verbal information in this context can be taken to include all sensations: visual images, sounds, touch, taste and smell (Eysenck 1990). According to Paivio's theory, verbal and non-verbal information are processed separately and have distinct representations within the brain. Thus, the image of a dog and the word 'dog' would be dealt with and stored quite independently. However, the two systems are connected so that we can, for instance, link our internal representations of the image and word for dog. Some experimental evidence supports Paivio's theory by suggesting that images and verbal information are processed in physically separate regions of the brain. People with localised brain injury can have quite independent deficits in either recognising or naming images. If the connection between the two brain hemispheres is cut, the language and imagery processing systems appear to work separately; subjects can recognise but not name objects, or vice versa. This suggests that the two systems are located in different hemispheres. Further evidence for Paivio's theory can be found in the fact that visual images are recalled more easily than words (Paivio 1971), possibly because memorisation of pictures involves the use of both systems; when we attempt to memorise a picture of a dog, for instance, we are likely to associate the image with the verbal concept 'dog'. Recall is improved when both systems operate. But when memorising words, we may be less likely to visualise what the word represents. Abstract words (such as 'justice') are remembered less well than concrete ones (such as 'a judge'). Abstract words are less evocative of visual images and so it is less likely that both systems will be brought into play when they are memorised. It is possible that recall is improved further because the use of images as well as words forces the individual to organise the information more—the increased amount of processing leading to an increased chance of successful recall (Avery and Baker 1990). Imagery may also be useful because it helps create associations between new material and existing general knowledge (Day and Bellezza 1983).

Structure of long-term memories

Long-term memory seems to encode the significance of events and experiences rather than their specific details. Many theories have been advanced to explain how our minds create meaning. Most concentrate on the ways that mental concepts and classifications are formed and the different ways in which they can be combined (Green 1996). Humans are naturally inclined to classify. Anthropological research on widely varying cultures shows a remarkable consistency in the ways that the world is divided into categories. This is not to say that every culture uses identical categories; Lakoff (1987), for example, discusses one that has a single category which incorporates women, fire and dangerous things! What all cultures have in common is the classification of experiences, and of the world generally, into a complex hierarchical collection of discrete and overlapping concepts. This ability seems to be innate, and the need to categorise seems to be a fundamental part of thinking. An example of a simple categorisation hierarchy is given in Figure B.3 (after Bower et al 1969).

Other evidence comes from research into problem solving, which shows that expert and novice problem-solvers approach problems quite differently (Best 1999). Experts seem to possess hierarchical mental categories into which problems can be fitted. Because of this they can structure information about a problem as it is gathered, which helps them to quickly choose the most appropriate method of solution. Frank Lloyd Wright observed that experts don't need to think because they *know* the answers to problems intuitively. Novices, on the other hand, have no such predefined framework, and so take much longer, or are unable, to structure the information they have been given about a problem. Because their knowledge is unstructured, choice of solution is more difficult or even impossible. Experiments have shown that information is much more easily remembered if presented in a hierarchical form (Bower et al 1969). This applies, for instance, to the presentation of educational materials, where the traditional 'linear' mode of delivery has been found to be less effective than a hierarchical presentation that mimics an expert's organisation of the material.

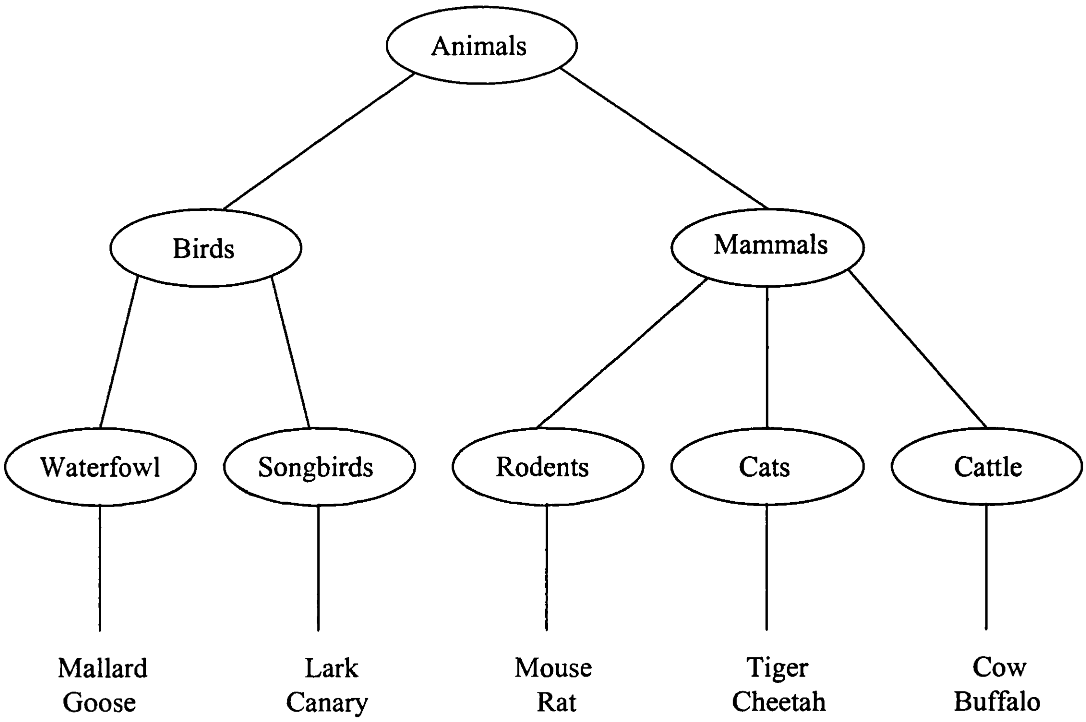


Figure B.3 Simple categorisation hierarchy.

Typically, the distinctions between categories seem to depend on attributes or characteristics of the members in each category. Collins and Quillian's hierarchical model of semantic memory (1969) allowed characteristics to be associated with categories. One example is given in Figure B.4. This model seems realistic because it shows that attributes are stored at the highest possible level of generality; we don't need to remember that a salmon can eat because we know that animals eat and that a salmon is an animal. Notice that internal contradiction is possible; one of the attributes of an ostrich (can't fly) overrides an attribute of a bird (can fly). The number of levels between concepts is a strong predictor of the time taken to verify statements (e.g. 'a shark has skin' involves three levels). This has been taken as evidence that Collins and Quillian's model might offer a good representation of how certain aspects of memory operate.

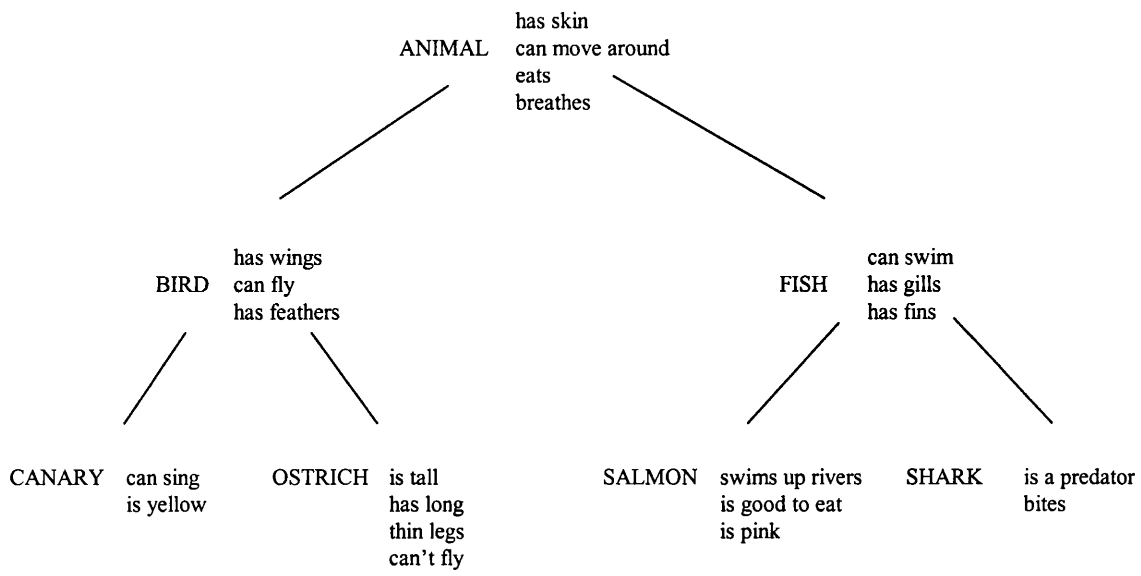


Figure B.4 Hierarchical model of semantic memory

It has been found that some concepts are not easily categorised, and some concepts appear to be more typical of their categories than other concepts. Additionally, some attributes seem to be more important than others in determining how typical a category member is. In response to these and other problems, the *spreading activation model* has been proposed, which represents memory as a semantic network of related concepts. The model can represent hierarchies: it includes *is a* links, to represent category membership, and *is not a* links to represent non-membership. No distinction is drawn between concepts and attributes. One example (from Best 1999) is given in Figure B.5.

The spreading activation model uses the idea of *semantic distance*; the further apart two related concepts are shown in the diagram, the more weak is the association between them. So, for example, 'yellow' is more closely associated with 'banana' than 'bus'. Only concepts that are directly related to one another are explicitly connected on the diagram; for example, 'bus' is not directly related to 'daffodil'. This model replicates quite well the experience of associative recall, in which thinking about one thing can make you think about related things. It shows how existing knowledge could be brought into play by the input of suitable cues.

What the spreading activation model fails to explain is how we are able to reapply knowledge of a more general nature (such as Schank's situational memory). Problem-solving by analogy involves the application of a general framework of principles, derived in one subject area, to a new subject area. It not clear how the spreading activation model of memory can deal with the wholesale re-application of such generic concepts. Later in this section we shall look at some theories that attempt to explain how we form and use such higher-level structures.

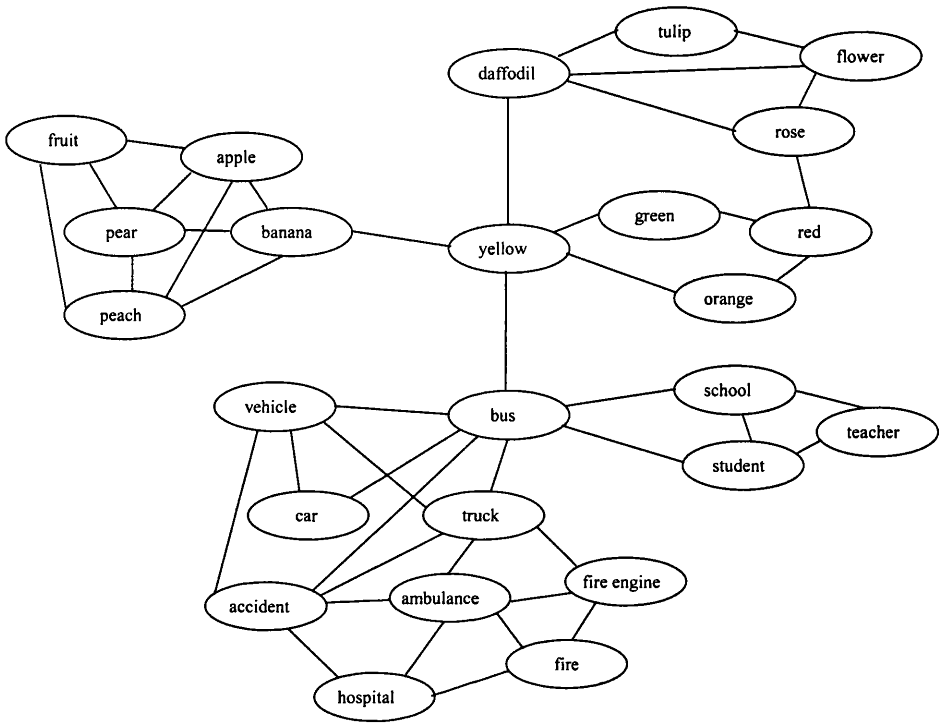


Figure B.5 Spreading activation model (Best 1999)

Short-term memory

Short-term memory is very different in nature from long-term memory. The role of short-term memory is to keep information at hand while it is of immediate use, either because the information is the result of recent sensory input, or because it has been retrieved from long-term memory. Information stored in short-term memory is directly available to the conscious mind without the need for appreciable cognitive effort. Short-term memory is used when we pay attention to, or think about, something. Just as the scope of one's attention is limited to a small amount of information at a time, so the scope of short-term memory is also limited. In his famous article on the limitations of human information processing capacity, Miller (1956) postulated that the mind could deal with only approximately seven (plus or minus two) items at any one time. The limit was later revised to three sets of three (Miller 1975). An important implication of this limitation is the constraint that it places on retrieval of information into short-term memory for the purposes of conscious manipulation. Conscious attention cannot be focused on a large number of individual items or concepts at once.

This restriction is a severe one that has obvious consequences in any tasks that require people to keep information in mind for extended periods. Short-term memory is used in language comprehension, and most people will have experienced at least one consequence of its limited nature: losing track of a long sentence before reaching its end. Sentences with many subordinate clauses require mental 'slots' to be taken up while the whole sentence is being processed. Johnson-Laird (1993) quotes the following sentence that requires additional storage in short-term memory:

The man the dog bit died.

The sentence requires a listener to hold its subject, *the man*, in memory whilst processing the embedded clause *the dog bit*, since the verb *died* does not occur until the end of the sentence. People with severe short-term memory deficits cannot deal easily with sentences of this type, because of the demands placed on memory 'slots'. The same information can be expressed in a way that does not place additional load on memory:

The dog bit the man and the man died.

Here no additional memory slots are required because the two clauses are in sequence rather than nested. Patients with damaged memory can interpret sentences of this type more easily. Experimental evidence using computers to parse natural language has shown that three slots are generally sufficient to deal with normal English (Marcus 1980). It should be understood that what each 'slot' represents need not be an elementary or atomic item of information; once a set of related ideas has been processed as a meaningful 'chunk' it need only occupy a single slot in memory. So, for example, the two ideas in the second sentence above, *the dog bit the man* and *the man died,* could occupy only two slots. Once the whole sentence had been comprehended as a single idea it might occupy only a single slot.

This principle of grouping items of information into meaningful sets and treating each set as a single unit is known as 'chunking' (Malim 1994). Chunking is necessary to help us cope with complexity. Short-term memory is extremely limited, and yet long-term memory is an almost unlimited collection of highly interconnected ideas. We are constantly bombarded by enormous amounts of incoming sensory information. One might think that we would never be able to focus attention on anything more than a tiny amount of this information at any one time. But people do manage to cope with large quantities of information, and chunking is one of the main mechanisms we have for achieving this. Although the capacity of short-term memory is limited to approximately three items at any one time, each of the three items, treated as a 'chunk', can refer to three more items. Each of those items can refer to a further three items, and so on, as illustrated in Figure B.6.

A common example of chunking in practice is the recall of long telephone numbers. The number '0035316081436' is unlikely to be remembered easily if treated simply as a series of digits. But it is more easily remembered if the digits are grouped (e.g. 00 353 1 608 1436). If meaning is assigned to each group of digits then remembering is even easier (00 is the international dialling code, 353 is the country code, 1 is the city code, 608 is the exchange and 1436 is the extension). And if any of these facts are already present in long-term memory, there is no need to remember them separately.

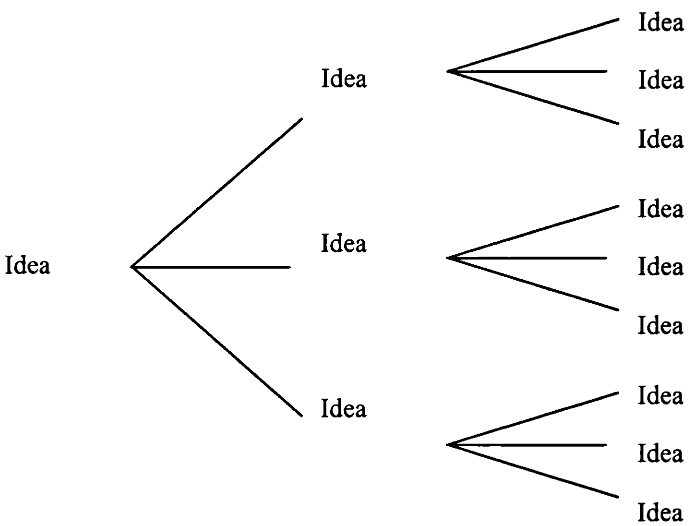


Figure B.6 Chunking

Another example of chunking is the use of abstract terms to denote complex ideas. It is often convenient to refer to whole areas of thought by a single name. The term *statistics*, for example, covers a range of concepts, theory, applications and assumptions. This kind of chunking can save an enormous amount of time and effort in communication, but depends entirely on the presence of a shared understanding of the terms that are used. For a demonstration read any academic article in an unfamiliar discipline!

Short-term memory is thought to use an *acoustic coding*, meaning that an auditory record of the sensory input is used. The capacity of short-term memory appears to be related more closely to the length of time taken to pronounce words than to the number of words to be remembered (Schweickert and Boruff 1986). The acoustic coding may explain why errors in recall from short-term memory tend to be 'transcription' errors such as replacing *b* with *v* and *x* with *s* (Malim 1994).

Strategies used to overcome the limitations of short-term memory do so by reducing the amount of information that must be held at any one time. For example, using a mnemonic allows several items to be remembered as a single unit. In the *method of loci*, originally described by Cicero, items to be remembered are associated to locations on a route, connecting the information into a single visualised journey in long-term memory. Any method that increases the power of short-term memory does so by augmenting it with other types of memory or by using artificial aids, as there is no evidence that the capacity or duration of an individual's short-term memory can be increased significantly. Short-term memory is inherently limited, which is something that those involved in the design of tasks with high cognitive content—such as the design and use of computer software—should take into account.

Various theories have been advanced about the mechanisms behind short-term memory. It has been likened to the working registers in a computer CPU since, unlike long-term memory, short-term memory is dynamic and its contents decay within seconds. Conscious repetition is necessary to retain information for long periods, an inefficient way of storing information that has been likened to *"using a helicopter to hold up a clothes line"* (Sowa 1984). Some psychologists think of short-term memory in terms of *working memory* (Baddeley 1986), which allows for separate processing of verbal and visual information. There is debate about whether visual images are held in short-term memory; experimental evidence suggests that a 'spatial' coding is used for visual information, so that we retain positional information that helps us to reconstruct an image rather than recalling the image itself (Malim 1994).

Normally considered to be closely related to short-term memory is *immediate* memory, which stores a visual or auditory image of sensory input for a very brief period before it is coded acoustically or spatially. Immediate memory allows us to continue to 'view' a scene for up to half a second after it has disappeared (*iconic* memory). Unprocessed auditory input remains available for up to four seconds (*echoic* memory).

B.3 Perception and comprehension

Perception refers to the processes by which we interpret sensory information received from the world, so as to build an internal view of what lies outside. Information from many sources around us is received via the senses of sight, hearing, taste, smell, touch and balance. The brain must co-ordinate and interpret all of this information to form a unified and seamless illusion of contact with the outside world. Perception and sensation are quite distinct processes:

> *"Perception is not determined simply by stimulus patterns; rather it is a dynamic searching for the best interpretation of the available data"*
>
> (Gregory 1966)

The mechanisms of perception are very effective. For many people, there is no question that the world is as they perceive it to be. Colours, sounds and smells are all present in the world, waiting for us to experience them. This view is of course incorrect. Colour does not exist except in the mind and is the brain's way of registering electromagnetic wave frequency. Music can be said to exist only because of the way our brain interprets certain types of oscillation in air pressure. Smell tells us only of the presence of particular molecules in the atmosphere around us. In other words, these sensations are mental analogues of real physical phenomena that are quite distinct from the sensations themselves.

Sometimes the nervous system has to work particularly hard to maintain the illusion of an objective outside reality. For example, when we walk outdoors the mix of light frequencies reaching our eyes changes significantly, but the colours of objects appear to stay the same. This effect is called *colour constancy*. If the colours that we perceived were a true reflection of the frequencies reaching our eyes then objects would change colour according to where they are. But our perceptual system automatically corrects the information that it receives so that we may continue to believe that things have certain unchanging colours regardless of their surroundings. Similar constancy effects are found in the visual perception of shapes and objects, and in the other senses.

Our window on 'reality' can never be more than an internally-constructed illusion. Most of us share a sufficiently common perception of the world to be able to act, for most of the time, as if the world did exist as our senses tell us. However, the specific mechanisms by which we experience the outside world remain something of a mystery. According to one early theory (Gregory 1972) perception is an active process which results in the creation of *perceptual hypotheses* about data received from the senses. The perceptual hypothesis is a 'first guess' about what incoming sensory data means, based upon experience. New information can cause us to automatically revise our perceptual hypotheses. In contrast, Gibson (1986) stressed the fact that incoming sensory data contains much encoded information that can be used in interpretation. According to his theory of *direct perception*, very little experience or higher-order processing is necessary to react appropriately to the perceived environment. Instead, the organism is able to glean sufficient information from attributes of incoming sensory data (such as depth cues, direction of reflections, parallax, and so on). Gibson's theory was a compelling one, but is generally felt to be inadequate. It is now widely accepted that much of perception involves interpretation of sensory input (or a transformed version of it) in the light of experience (Eysenck and Keane 1995). This alternative view is known as the *constructivist* theory of perception (Best 1999).

In more recent years the mechanics of vision have begun to be explained in a convincing way. David Marr's pioneering work (1982) on what is termed 'early vision'—the earliest stages in the processing of incoming visual stimuli—described the *computational* approach to vision. In Marr's view the brain, and its associated neural pathways, transform and interpret visual stimuli by carrying out computations upon them. *Connectionist* research, using computer simulation of brain-like neural networks, has tended to confirm his view. Our ability to recognise shapes seems to emerge as a direct consequence of the way in which signals are transformed by the retina, optic nerve and visual cortex. One of Marr's major contributions was to demonstrate that the retina is capable of performing these transformations. No reflection need therefore be involved in identifying basic shapes or even in recognising more complex objects such as faces. Once childhood experience has 'trained' the relevant parts of our brain to process signals correctly—to recognise and differentiate between them—the process is automatic and mechanical. It is not the product of 'thought' in any conventional sense, whether conscious or otherwise.

Pattern recognition and classification

One of the most fundamental tasks that our perceptual system must perform is to correctly identify patterns in incoming sensory information. Without this ability we would be unable to recognise any objects in the world around us. The Gestalt psychologists identified several principles of perceptual organisation that are fundamental to pattern recognition, although they did not offer any explanation of the perceptual processes underlying these principles (Eysenck and Keane 1995). They observed that we have the innate capacity to organise our perceptions in meaningful ways, and expressed our capacities as 'laws' of perception, which are illustrated below in visual terms (from Malim 1994). The Gestalt idea applies not only to recognition of pictures, but to any visual recognition task, such as reading.

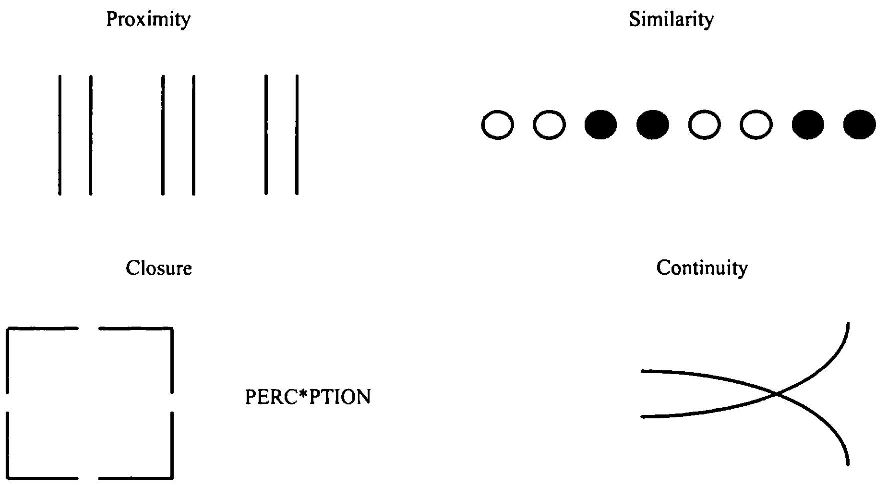


Figure B.7 Gestalt laws of perception

Proximity Elements close together are perceived as belonging together. The six lines in Figure B.7 are perceived as three groups of two, because of the proximity of the lines in each pair.

Similarity Elements which look similar are mentally grouped together. The circles in Figure B.7 are perceived as pairs because of their visual similarity, even though all of the circles are equally spaced.

Closure Incomplete figures tend to be perceived as complete ones. The broken shape in Figure B.7 is perceived as a square, even though we can see quite easily that it is not a complete square. In a similar way, the letter missing from the word 'perception' is automatically filled in.

Continuity Figures defined by a single unbroken line tend to be seen as an entity. The example in Figure B.7 is perceived as two crossing lines rather than two pointed shapes, even though it could equally well be either.

A range of theories attempt to explain the mechanisms underlying recognition of visual images. One theory is that we have in long-term memory a number of template objects so that, when we see an object, we can identify it only if it matches one of the templates. For example, we could have several templates for each letter of the alphabet (e.g. A, a, α, A, a, 𝒜, ...). The main problem with this approach is that we would need a huge number of templates even for simple objects such as letters. We would also not be able to recognise parts of shapes or new variations of known shapes. It is more likely that we use a feature matching approach, in which the constituent parts of a picture are individually matched. A picture is recognised if its features correspond closely to those of a picture held in memory. If several 'matches' are obtained, the closest one is used. Experimental evidence confirms this view; even when information is ambiguous our perceptual processes tend to resolve the ambiguity automatically, and we may well be unaware that multiple interpretations are possible. Experiments with the recognition of spoken letters show that intermediate sounds are heard either as one letter or as another, with no middle ground (reproducing the sounds of a foreign language can be difficult for this reason, since one may not even be able to hear them). Well-known visual illusions that can be interpreted in several different ways demonstrate the same effect—we can perceive one interpretation or the other, but never both simultaneously.

One of the most important aspects of any recognition task is *context*. What we perceive depends heavily on the context in which the perceived object is placed. For example, one's ability to read is not heavily impaired even if l-tt-rs are m-ss-g from s-me w-rds. In the following two sentences, the powerful effects of context cause us to recognise characters in unusual ways:

THE 13OY IS 7 YEARS OLD
HI5 NAME IS 5TEVEN

Verbal information can act as a context for visual information, and vice versa. Visual recognition is slightly improved if textual cues are given, and verbal recognition is significantly improved if visual clues are given (Avery and Baker 1990). One reason for these effects is the context provided by the additional information, which reduces the range of potential interpretations, and helps us to home in on the correct interpretation.

Mechanisms underlying pattern recognition

Software simulations of the brain's neural networks can recognise patterns such as faces, provided that the software networks have first been 'trained' appropriately. Training proceeds by a process of trial and error in which correct responses by the network are reinforced; eventually, the network 'learns' to give the desired responses to incoming signals. The fact that no algorithmic programming is required increases the biological plausibility of the neural network model. Pattern recognition happens very quickly—far more rapidly than would be possible if any form of linear searching were taking place. The speed of both visual pattern matching (e.g. reading and face recognition) and auditory pattern matching (e.g. interpretation of speech) suggests that very low-level processes are involved. This evidence also supports the connectionist model. It can be shown that the operation of a neural network is equivalent to a series of rather complex mathematical calculations. For instance, some networks recognise patterns by effectively calculating Fourier transforms of them. A neural network can therefore be viewed as a form of analogue computer. The brain is not a single network of interconnected neurons, however. It contains many distinct subsystems that appear to perform more or less specialised roles. Many cognitive actions (such as speech) are handled simultaneously but differently in several regions of the brain (Johnson-Laird 1994). And specific parts of the brain have overlapping functions; the visual cortex, for example, deals not only with vision but also with hearing (Barlow 1990). After brain damage, remaining parts of the brain can sometimes take on the roles of destroyed parts. Artificial neural networks have yet to duplicate the brain's higher-level structure.

We have seen that much recognition, especially for basic shapes and images, occurs more or less automatically and without the need for conscious thought. The extent to which this is true of higher-level perceptions is less clear. Significant analytical mental effort may be needed to recognise complex and ambiguous scenes, and this process may even be sufficiently conscious to take the form of a verbalised (though not necessarily vocalised) train of thought. But there is a large middle range of visual recognition tasks where recognition is essentially automatic. The neural network model of recognition may go some way towards explaining why we are capable of absorbing visual information so easily. Visual recognition by-passes the slower verbal and analytical modes of thought. Despite our feeling of conscious control, the physiological signature of recognition in the brain is observable well before we become consciously aware that recognition has occurred. Visual presentation of information offers one way of communicating meaning directly without having to rely on conscious processes. Even some higher-order decision-making seems to occur automatically, before we have conscious knowledge of our own choices. Many people have observed that they don't know what they think until they hear themselves say it! It seems that most cognitive activity is unconscious, and consciousness must lag behind, reflecting a small part of what occurs subconsciously. Consciousness is perhaps like a screen on which a selective view of the 'real' action is projected.

Attention

In order to deal with the multitude of sensory inputs that we constantly receive, we must be able to devote attention selectively. Selection of the focus of attention is often automatic; for instance, it is difficult not to be distracted by loud, colourful, moving images (as on a television screen). At the same time, we can discriminate between a variety of inputs and concentrate so effectively on a selected thread that we may even be unaware of other inputs. Early theories of attention proposed that the unwanted sensory input is discarded before processing in some form of sensory buffer or *filter*, allowing the brain to process the wanted input without becoming overloaded. However, it has been observed that, while devoting attention to one thread of input, we are still able to detect relevant or interesting information in unattended threads. This is sometimes called the *cocktail party phenomenon*; people engaged in a particular conversation can somehow manage to pick up 'interesting' information (such as a name) from many other conversations elsewhere in the room. While this may seem unremarkable, in fact it implies that the information we apparently ignore must actually be processed at a similar level to that of attended material. This is because we can determine its relevance only on the basis of its meaning. The only real difference in level of processing between attended and unattended material two seems to be that attended material is passed to the conscious mind (Kellogg 1997).

These findings imply that the brain does much more work than conscious thought alone would imply. The two types of processing—conscious and unconscious—have respectively been called *controlled* and *automatic*. Controlled processing is the sequential, conscious thought required by difficult and unfamiliar tasks. Automatic processing is the rapid, effortless mode of thought that becomes possible when a task is simple and familiar, and which drastically reduces the time taken to deal with well-understood tasks. For example, some people can identify spelling mistakes in a page of text without having to read each word. This can only happen because the mind is able to use automatic processing to read the page unconsciously. In autistic people, this kind of ability can be so well developed that subjects have been reported who could 'count' large numbers of objects accurately at a glance—much faster than is possible using conscious, step-by-step counting methods (the reason why these skills should be so well developed in autistic people is unknown). Treisman and Gelade (1980) showed that automatic processing (which they called *pre-attentive processing*) can deal with many items at once, provided that only isolated features are sought and that the target items differ from the surrounding irrelevant items in colour, size or orientation. If combinations of features are sought, then automatic processing cannot be employed and the slower, sequential, processing (which he called *focused attention*) must be used instead.

The brain also uses parallel processing in the separate handling of each sense. People can process visual and verbal information independently, with relatively little interference between the two types of information. This means that it is possible to increase one's 'attentive bandwidth' by receiving verbal and visual information simultaneously (Shaffer 1975).

Comprehension

Comprehension is the process of interpreting and assigning meaning to the sensory input we receive from the world. Although we may talk about perception and comprehension as distinct activities, they are in fact inseparable and can be considered as two sides of the same coin. The end result of comprehension is a mental model (see next section).

The ability to comprehend language is affected strongly by choice of vocabulary. Even in standard English there are many ways of saying the same thing. A statement can use short words of Anglo-Saxon origin or longer, more 'refined' terms that English has acquired largely from French (Burgess 1992). In the UK, social class has historically been a strong determinant of whether an individual uses restricted or elaborated codes of language. Regional dialects have their own vocabularies. Using the 'wrong' vocabulary can cause difficulty. Studies of language comprehension show that choice of dialect is one of the most important factors affecting understanding. The performance in IQ tests of schoolchildren improved drastically when tests were written in their own dialect of English (Miele 1979). Predictably, children who did not use the same dialects performed poorly in the modified tests.

The form of a message is also important. According to Chomsky (1968), language has a *deep structure*—equivalent to its irreducible meaning—and a *surface structure*, its form when written or spoken. Equivalent but non-identical statements have the same deep structure but different surface structures. Chomsky claimed that statements couched in a form closest to the deep structure are easiest to understand. For instance, *The bull chased John* is close to the deep structure, whereas *Wasn't John chased by the bull?* involves several transformations and is consequently more difficult to interpret.

The layout of information on a page or screen can have a significant impact on comprehension, as graphic designers know. Perhaps what makes an aesthetically pleasing layout also maximises comprehension and transfer of information. Bellezza (1983) found that distinctive page layout improved learning. A distinctive layout helps associate the items it contains (Bellezza calls this the 'spatial arrangement mnemonic') and may itself convey useful information.

Memory is necessary for comprehension. Short-term memory stores the intermediate results of processing as well as unprocessed sensory input. Long-term memory provides meaning, since language can have no meaning unless words can be linked to known concepts. Many theorists have suggested that conceptual information and lexical information are stored separately and organised differently within the brain (Kellogg 1997). This would account for the fact that people with brain damage can lose the ability to name objects, and yet still know what the objects are and how to deal appropriately with them (such people are called *aphasics*). Similarly, most people have experienced the "tip-of-the-tongue" phenomenon, in which we are aware that we know an object's name but cannot recall the name.

Anything that improves recall is likely also to improve comprehension. Recall of concepts and hence comprehension itself are dependent on physical context—it is easier to think about something if we are in the same surroundings as when we first thought about it. Memory is also improved if information is meaningful, since this allows a structure to be given to the information (Medin and Ross 1997). Hierarchical presentation has been found to improve recall. Autobiographical information is more easily retained than other information (Thompson et al 1987) and is therefore more easily understood, possibly because personal events are inherently more interesting or perhaps because of the additional processing that memories of personal experiences undergo.

Arousal is a term used by psychologists to describe the level of activity within an individual's brain. In rough terms, we could say that the level of arousal corresponds to the individual's degree of alertness. Vigilance, the ability to sustain attention, is directly related to the level of arousal. Increased arousal is likely to lead to improved comprehension. Arousal and vigilance are both known to be increased by stimulating factors such as the occurrence of surprising or novel events, use of certain drugs, movement, and the presence of noise or bright lights. They are decreased by monotonous or boring tasks and repetitive stimuli. If arousal increases up to a certain threshold, the optimum performance level is reached and, beyond this, performance decreases with further increase in arousal level.

An individual's frame of mind can also influence their ability to concentrate, understand material, and retain this information in memory. This factor is encapsulated in the term *set*, which can describe a range of emotional, motivational, social and cultural predispositions. The effects of set include a readiness to respond to particular signals, bias in favour of certain signals, and interpretation of particular signals before they occur (Malim 1994). A set is an expectation; the individual is ready to respond to some inputs but not to others. Sets can affect perception dramatically; as in the well-known saying 'you see what you want to see'. In one study, white South Africans were found to be unable to differentiate between faces from non-white racial groups when they were presented very briefly. Non-South African whites were able to differentiate successfully. A cultural set influenced the South Africans' perceptual processes and rendered them incapable of noticing clear differences in appearance. Other studies have shown that many individuals have a cultural set against 'taboo' words, which manifests itself in a measurable inability to *perceive* the words as effectively as non-taboo words. Many other types of set can affect an individual's ability to deal with particular topics. Some can decrease our problem-solving abilities quite significantly. Malim (1994) describes the following types of set.

Type of set	**Effects**
Set of operation	An assumption that a problem will be solved by means of a particular operation or set of operations.
Set of function	An assumption that things have a fixed function.
Set of rule	An assumption that there are certain rules, within the constraints of which the problem will have to be solved.

Table B.3 Types of set

B.4 Mental models

A *mental model* is an individual's internal representation of a situation (Manktelow and Jones 1987). Visual metaphors like 'mind's eye' and 'insight' can be misleading because a mental model need not necessarily involve any inner visual representation. A mental model is perhaps better thought of as a set of beliefs about a situation that provides a framework for thinking and can help us rationalise and predict. Humans are instinctive 'hypothesis builders' and use mental models all the time. When we observe and interpret the actions of another person we are likely to construct a mental model to help understand and predict their behaviour. When we come into contact with organisations we inevitably form mental models of how they operate so that we know how to deal with them. Construction of mental models is spontaneous, involuntary and unconscious.

Mental models are by definition incomplete and may be inconsistent and incorrect (Johnson-Laird 1985). They tend to mimic observed reality but may distort or misrepresent it (Eysenck and Keane 1995). Incorrect mental models cause us to act inappropriately. When our mental models cease to reflect reality (and we become sufficiently aware that this is the case) we experience 'cognitive dissonance' and may try to revise them so that they once again provide adequate explanations of reality. But, typically, people can be very resistant to changing their mental models, preferring instead to develop elaborate explanations that remain within the bounds of their current models (Johnson-Laird 1985). Children often produce erroneous explanations for the behaviour of the adult world, which implies a deficiency in their mental models. Adults may do the same thing when confronted with complex regulations or procedures (such as tax law or computer systems), even invoking the supernatural if it seems necessary. A mental model may be wrong in detail, but can nevertheless appear to 'explain' and even to predict correctly under certain circumstances. A model that apparently explains a situation well enough to be useful is likely to be considered adequate, even if it is in fact incorrect or incomplete. In other words, mental models are no more complex than they need to be (Norman 1983).

One example of the predictive use of mental models in everyday life is in the perception of relationships between events. If two events are temporally related (occur at the same time) then the mind is likely to posit a causal relationship between them (Kolodner and Reisbeck 1986). It is perceived that the events share the same cause, or that one causes the other, even if no such relationship exists. *"Red sky at night, shepherd's delight"* is a perceived relationship which happens to reflect a genuine causal relationship. Other perceived relationships (such as the seven years of bad luck that follows breaking a mirror) do not reflect genuine causal relationships.

One of the difficulties of theorising about mental models is the almost infinite variety of concepts that they can contain. Someone's mental model of even a relatively restricted area, such as their job, could include concepts as diverse as people, actions, business and social relationships, products, materials, business processes and tasks, transactions such as sales, payments and contracts, countries, companies, government departments, other organisations, rules, procedures, norms, responsibilities, levels of authority, lines of demarcation, places of work, other locations, documents and information in any number of forms, knowledge and abstract ideas such as 'quality', equipment, buildings and other physical objects of all kinds, personal goals, likes and dislikes, hopes, fears, prejudices, problems, solutions, animosities, rivalries, agendas (hidden or otherwise), health, mental states, attitudes, and a host of other concepts. In fact, anything you can think about can be part of a mental model!

Types of mental model

The mental models we use to understand the world are conscious structures derived from concepts in long-term memory. Theories about mental models are therefore essentially theories of meaning, since a mental model creates meaning by linking known concepts. If we can understand how mental models are internally represented then this may give us clues about how the mind structures knowledge. Figure B.8 summarises different types of knowledge structure that have been proposed (Eysenck and Keane 1995).

The assumption underlying most research into mental models is that they are made up of concepts and the relationships between concepts. We have already discussed some ways in which simple objects and relations between them might be represented in long-term memory. Eysenck's diagram lists some of the more complex types of knowledge structure that have been proposed: schemata, frames and scripts. In addition to these, we will be discussing later in this appendix memory organisation packets, thematic organisation points and thematic abstraction units.

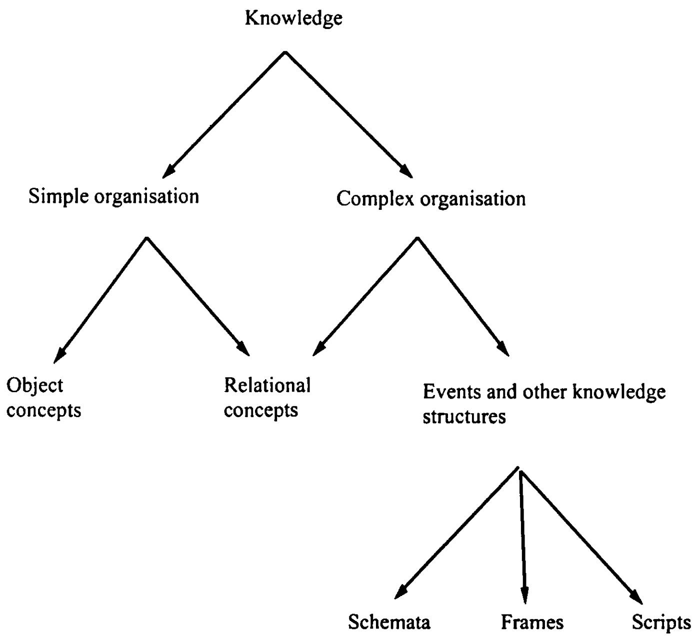


Figure B.8 Taxonomy of internal knowledge representations (Eysenck and Keane 1995)

Johnson-Laird (1993) groups proposed types of mental model into three distinct categories: images, propositional networks, and analogical mental models. Images are visual representations ("the mind's eye"). Propositional networks include the hierarchical classification schemes and semantic networks described previously. Analogical mental models are internal representations that are analogues of the situations they describe. A good way of distinguishing between the three types is to imagine different mental models of a house. A mental model in image form could be a photographic visualisation of a house in front elevation. A propositional network might describe the house with a series of statements such as 'There are two windows at ground level, with a red door between them'. An analogical model could be a stylised, imaginary 'scale model' of the house that one could mentally walk around: opening doors, switching on lights, eating food from the refrigerator, and so on. According to Johnson-Laird (1985) the first two types of model are not true mental models because they fail to mimic the structure of reality—they are not *isomorphic* to the world they represent. However, the consensus seems to be that real mental models must have some of the properties of each type of model. The analogical type of model seems to be of a higher order than the other two, so perhaps we should think of analogical models as conscious manifestations of unconscious information held in memory. In other words, the mind's internal working may use images and propositions to create conscious analogical models on demand. 'Photographic' images alone would not be sufficient as mental models because of their lack of internal structure. There is debate over whether images are stored in memory or reconstituted as needed (Eysenck and Keane 1995). If images are reconstructed then the lower-level information used to reconstruct them may well be used in the formation of mental models.

Concept formation

One view of knowledge has it that information is represented within the mind as a set of linked facts (or *propositions*) which together form a semantic network. Different types of network have been proposed, but all share the idea that mental concepts and associations between them form coherent conceptual structures. A well-known model for conceptual structures is Sowa's conceptual graphs (1984, see later). Some mental concepts, such as movement, are probably innate, while others are learned. Schank (1985) has proposed that a wide variety of ideas can be reduced to combinations of a small set of innate conceptual primitives, listed in Table B.4. A compelling aspect of Schank's theory is the idea that complex concepts are made up from simple ones, and this hints at a possible explanation for the mechanisms underlying 'meaning'—that we in effect reduce every experience into a combination of basic percepts or innate conceptual primitives.

To propel	To change location
To move a body part	To change an abstract relation (e.g. possession)
To expel	To create a thought
To grasp	To construct new information
To speak	To attend

Table B.4 Schank's conceptual primitives

The ability (and need) to categorise is fundamental to thought. Every concept, whether learned or innate, can be placed in one or more mental classification schemes. This ability lets us generalise (e.g. all animals have skin) and deduce (e.g. salmon have skin because animals have skin and salmon are animals). As a result, we can store and represent information with some economy. People from every culture make extensive use of classification. But it is unclear how we arrive at categories and how we decide what category any given object belongs to. In theory, categories can be defined in two ways: intensionally and extensionally. An intensional definition is an abstract one that states the rules for membership of the category; 'all people with red shoes on' is one example. An extensional definition is a list of the category members. Some categories defy definition in one or the other way; the category 'member of NATO' would be difficult to define adequately except by listing individual NATO members. On the other hand, it would be practically impossible to define the category 'grain of sand' in an extensional way. Wittgenstein used the idea of a game to demonstrate that some concepts are effectively impossible to define intensionally; any definition of the concept 'game' will exclude certain activities generally accepted to be games. This lack of clear definition is apparent in many concepts when afforded close scrutiny.

Concept definitions can vary according to context. Eysenck and Keane (1995) refer to this effect as *concept instability* and cite the example of the concept 'piano', which can mean slightly different things in the contexts of piano tuning and furniture removals. The underlying concept of a piano does not change, but the most salient attribute (ability to produce music and great weight, respectively) does. Experiments have shown that the most appropriate cues for a concept are heavily dependent on the context in which the concept is currently associated.

When we classify objects into categories we unconsciously rely more heavily on some attributes than on others (Malim 1994). Additionally, some objects seem to be more 'typical' of their categories than other objects. A robin and a stork are both birds, and yet experiments have shown that people do feel that, in some sense, robins are 'truer' birds than storks. People can readily rank objects as more or less typical of their categories, and the objects ranked as more typical turn out also to be more easily recognised (Best 1999). A fully extensional definition for many types of mental category would be impractical. But the evidence for 'fuzzy' and unstable categories, typicality and attribute salience points to ways of defining categories that are not entirely intensional. One theory proposes that categories are composed of a conceptual core of defining features together with a set of other characteristic features that can be used in identification (Medin 1989). Another theory suggests that prototype or exemplar objects define categories on the basis of similarity; the more an object is like the mental prototype, the more 'typical' it is seen as (Eysenck and Keane 1995). The common prototype for a bird would presumably be more like a robin than a stork. The prototypical object for the concept 'furniture' might be a chair or table; one might have no difficulty in classifying a desk as furniture but not be so sure about a mirror. A criticism of this theory is that it is perhaps rather difficult to think of adequate prototype objects for certain concepts, especially abstract ones like *integrity, justice* or *God.*

It has been observed that people tend to be more comfortable with certain levels of generality than with others (Rosch 1975). This optimum level of generality seems to coincide with everyday experience—the preferred categories are concrete ones that apply to everyday objects. For example, the concept 'chair' is an everyday one for most western people whereas the more general category 'furniture' is too abstract to be easily describable except in very general terms. A more specialised concept, such as 'wooden chair', would have few unique attributes that a chair would not. Eysenck and Keane (1995) refers to these levels as the superordinate (general), basic, and subordinate (specific) respectively. Other evidence (Lachman and Butterfield 1979) confirms the difficulty of dealing with possibly unfamiliar and vague generic categories (one example cited is 'mammal').

Propositional networks

Many types of propositional network have been formulated in an attempt to explain how concepts are related to create meaning. We have already seen semantic networks (see discussion on long-term memory, above). For Sowa, a conceptual structure (1984) is an abstract and rigorous statement in which concepts are related to, and defined in terms of, other concepts. Sowa's conceptual structures were developed to help explain how we construct mental models, but also to point the way for the construction of artificial intelligence. The conceptual graph in Figure B.9 represents the sentence *The girl Sue is eating a pie fast.*

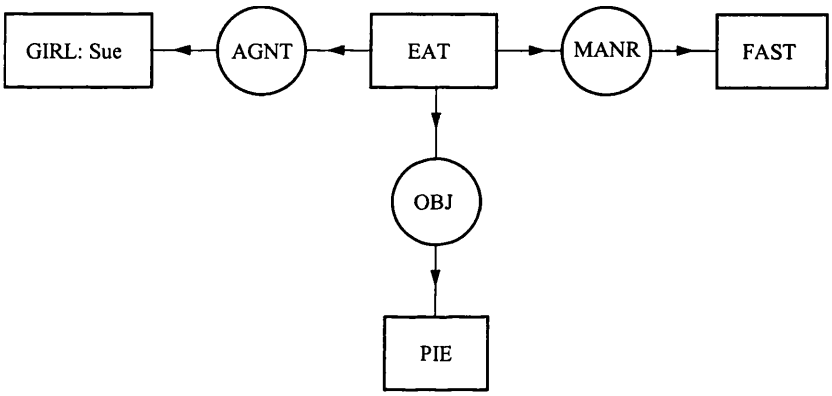


Figure B.9 Simple conceptual graph (Sowa 1984)

In this graph, the concepts [GIRL: Sue], [EAT], [PIE], and [FAST] are associated via the conceptual relations (AGNT), (MANR) and (OBJ). The concepts should be self-explanatory, with the exception perhaps of [GIRL: Sue], which refers to a particular instance (Sue) of the general class [GIRL]. The conceptual relation (AGNT), between [GIRL: Sue] and [EAT], tells us that it is Sue who is doing the eating (in Sowa's words a particular girl, Sue, is the agent responsible for an instance of the action [EAT]). The conceptual relation (MANR) says that the manner in which the eating is being conducted is [FAST]. The relation (OBJ) indicates that the object of the eating is a [PIE].

Sowa's approach includes extensions to cater for reasoning and deduction, representation of a range of language constructs and the design of information systems. One of his most important contributions was in showing how higher-level concepts can be constructed using lower-level ones. For example, the concept [BUY] can be represented as in Figure B.10.

Type BUY(x) is

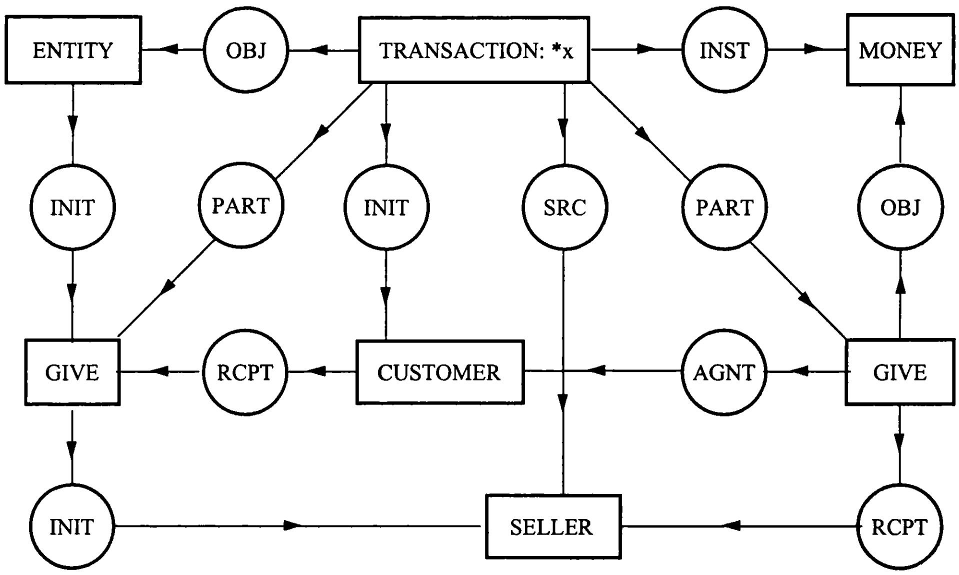


Figure B.10 Conceptual graph for concept BUY

This graph gives us a definition for the concept BUY: the transaction is initiated by a CUSTOMER who is the recipient and agent respectively of two acts of GIVE. The transaction's object is an ENTITY, which is also the object of the first GIVE. MONEY is both the instrument of the transaction and the object of the second GIVE. A SELLER is the source of the transaction, the agent of one GIVE and the recipient of another. Having defined the concept BUY, we may make use of it. The following graph depicts *Joe buying a necktie from Hal for $10* (Sowa 1984).

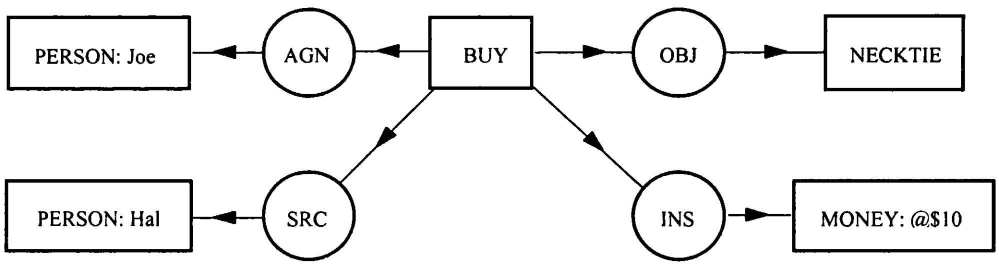


Figure B.11 Concept BUY in use

Other mental structures

One proposed large-scale memory structure is the *schema*, a stereotyped set of knowledge about particular procedures, sequences of events or social situations: "*an organised configuration of knowledge derived from past experience, which is used to interpret our current experience*" (Avery and Baker 1990, Singer and Kolligan 1987). A schema provides expectations about what should happen in situations of a given type, and helps us make predictions about related types of situation. The earlier discussion on the contents of long-term memory introduced the idea of a *script*, a schema that refers to a specific situation and includes a standard sequence of events (Eysenck and Keane 1995). A script can be thought of as a series of scenes depicting the stages that generally occur in a particular type of situation (c.f. Schank's situational memory). For instance, an individual might have a script that covered visiting the doctor. The script would contain scenes for parking a car, talking to a receptionist, waiting to see the doctor, explaining to the doctor what the problem is, being examined, and so on. Scripts are generalisable, reflecting the fact that we can apply past knowledge when in new or unfamiliar situations. The 'doctor' script might be helpful in predicting what sort of things happen when visiting, say, a vet for the first time. On the other hand, using the script could be rather inappropriate in situations such as visiting an accountant, and we might find that applying it leads to problems. In this case, we may be forced to revise our script or create a new one.

Stereotyped views about certain types or groups of people are also examples of schemata. Often, our schemata are not particularly helpful, since they force us to view a situation in a particular way. Racial prejudice is one example of the results of unhelpful schemata. The phenomenon known as 'paradigm shift', when someone appreciates a situation in an entirely new light, can be explained in terms of the replacement of one schema by another.

Schank (1985) developed the script concept further with the idea of *memory organisation packets*. Whereas a script is a linear sequence of scenes, a memory organisation packet is a hierarchical script that allows substitution of scenes when appropriate. The ability of memory organisation packet to be modified through the replacement of scenes reflects our own abilities to use and adapt past experience in new situations. Without the ability to respond flexibly to circumstances we would find life extremely difficult. To add further generalisation capabilities to the memory organisation packet model, Schank proposed that *thematic organisation points* are also retained in memory. Thematic organisation points are highly generic 'storylines'. For instance, the story of Romeo and Juliet could be classified using a thematic organisation point for 'mutual goal pursuit against opposition'. The fact that thematic organisation points are so generalised makes them appropriate for problem-solving by analogy. People can transfer experience from one area of life to another but, surprisingly, evidence suggests that they do this only when consciously trying to. In other words, we do not automatically apply knowledge that might be useful, even if the broad structure of a situation (such as the mutual goal pursuit example above) is essentially the same. This implies that thematic organisation points, or something like them, do exist, but are applied only when a subject consciously searches for an appropriate match, either because they remember to do so or because they are asked to.

Dyer's *thematic abstraction units* act in a similar way to Schank's thematic organisation points. Dyer characterises thematic abstraction units as embodying the kind of knowledge that we pass on in the form of proverbs, such as 'A stitch in time saves nine'. Although we might not take proverbs seriously, experimental evidence suggests that people do actually apply heuristics of this nature in general problem-solving tasks, even (or especially) when they have no specific knowledge of the situation on which to base their thinking.

Connectionist view of mental structures

The connectionist view explains mental processing rather differently: in terms of the mutual effects of networks of interacting neurons. According to this view, the observable properties of the mind emerge from the combined action of groups of neurons. We have already seen how visual recognition occurs because the arrays of neurons lying between the retina and visual cortex process signals automatically. For example, the brain's visual feature detection systems (retina, optic nerve and visual cortex) could recognise the fact that a scene contains a moving object and that it is grey and furry. Processing of incoming auditory signals could recognise the noise as a purring sound. The neural network that receives all of these inputs—motion, greyness, furry texture, purring sound, and so on, together with context—would recall memories based on this specific combination of features. One of these memories is likely to be the word 'cat'. Other memories are likely to be of other cats, or experiences that were associated in some way with moving grey fur and purring sounds. These memories would be consciously experienced and hence the observer would experience some meaning, which we could label as 'cat'. The concept 'cat' is thus an *emergent* property of the retrieval mechanism. It is not stored as a discrete concept in the brain.

According to this (connectionist) perspective, if I think of the word 'cat', I do not locate the concept *cat* in a semantic network and then explore nearby linked concepts. Instead, I involuntarily recall perceptions I have associated with the word 'cat' through previous experience. These associated sensations—such as the feel and look of a cat's fur—are triggered and re-experienced in my mind when I imagine myself experiencing the sound of the word 'cat', and consequently I re-experience the sensations, or a form of them: this is meaning. No concepts are involved. Assuming that I have been exposed to other animals, then the sensations that I recall are likely to trigger the recall of similar sensations to do with other animals (e.g. ones that have fur, or have four legs, or have teeth) and so I am able to generalise the concept of *cat* into the wider idea of an *animal* (essentially, 'things that remind me of cats'). But nowhere in my mind is the concept *animal* stored; what we call the concepts *cat* or *animal* are simply labels that we assign to the collections of recalled perceptions that we experience when prompted with the words or other suitable cues.

Without going into the complexities of brain structure, this is a biologically plausible explanation. No physiological evidence has been found of any brain structures that correspond to schemata, scripts, concepts, thematic abstraction units, thematic organisation points, memory organisation packet or any other of the proposed higher-order structures. But it has been demonstrated through simulation that simple networks of neurons can perform the kinds of tasks described above (Rumelhart and McClelland 1986).

One difficulty with semantic network theories is that they lead us to think that concepts exist somewhere in the brain, embedded in a network of associations. If that were true then the semantic network would need to be interpreted in order to be understood. This is the *homunculus problem*: there is no inner person within the mind to do the interpretation, and, if there were, how would their own mind work? It is probably more accurate to regard semantic networks as no more than an attempt to explain the observed behaviour of the mind, that does not reflect its actual structure (Solso 1998). The nodes in semantic networks denote discrete concepts. In contrast, the nodes in neural networks have no inherent content. Content in a neural network is held throughout the network, not in individual nodes. According to this view, thinking is the experience of sensations, both immediate and remembered—reliving and recombining experiences and sensations we have received via our senses. Memories are stored sensory perceptions (images, sensations of movement, smells and sounds, emotional states, etc.). Associative recall is triggered either by external sensory input or by thoughts, which in turn trigger the recall of other perceptions. Concepts emerge during recall, through the concerted action of the brain's neurons.

This inherently 'fuzzy' way of storing information has been referred to as *non-propositional representation of knowledge* (Bechtel and Abrahamsen 1991). Knowledge is distributed throughout the nodes and their connections, and can be retrieved only through the concerted action of sets of nodes. Holograms offer a good analogy for this effect—since a holographic image is an emergent property of the *whole* hologram. One cannot locate any part of an image in a hologram. Damage to specific parts of the hologram results only in general degradation of the whole image. This similarity has led some to suggest (not very accurately) that the brain works 'holographically'. This explanation of thought is only one current theory and remains to be proved or disproved. But it is a compelling theory, because it avoids the need for any internal semantic network. It suggests that concepts come about through associations between sensations recorded in memory.

Analogical mental models

Analogical mental models are different in kind from other types of mental model in that they are analogues of the situations they describe. An analogical model encodes the same kind of knowledge that a propositional network can encode, but does so in such a way that details are hidden (Johnson-

Laird 1993). When we visualise a house, for example, we do not need to think about the logic inherent in its structure. This is similar to the way in which our perceptual system carries out complex transformations on incoming sensory signals, but presents the final result to consciousness in analogue form. The end result of the perceptual process—a visual image of a person, for instance—is so real to us that most people remain unaware that the mind has to work hard to create an illusion of direct contact with reality. In a similar way, our mental models are typically indistinguishable from reality, and for much of the time this assumption does not cause us problems.

Analogical mental models are concrete, conscious visualisations which are isomorphic to (i.e. they have the same form or composition as) the situations they represent (Johnson-Laird 1985). They are not abstract statements of fact. A mental model 'looks like' the real thing and may function like it, although there is nothing to stop us breaking conventions or defying natural laws such as gravity within a mental model. A mental model is like a working model of a real situation. For example, we may have a mental model of how a car works. This could include a visualisation of a car, complete with wheels and engine. If we know something about how the engine works, then the mental model may extend to components of the engine itself, and we might envisage a carburettor feeding fuel to cylinders, a distributor providing electricity from a battery to spark plugs, and so on. However, at some level of detail our knowledge inevitably runs out, and at this point the mental model can no longer reflect reality. For example, we may have no model at all (or an incorrect one) of the internal functioning of the carburettor.

In consideration of the rents and the covenants on the part of the Lessee and the conditions hereinafter reserved and contained the Lessor HEREBY DEMISES unto the Lessee (a) ALL THAT the premises more particularly described in the First Schedule hereto all of which premises are hereinafter referred to as "the Demised Premises" (being part of the land buildings and appurtenances at (address omitted) such land buildings and appurtenances being edged blue on the plan annexed hereto and being hereinafter as "the Development") TOGETHER WITH the appurtenances thereunto belonging AND TOGETHER ALSO with the several easements and other rights specified in the second Schedule hereto EXCEPTED AND RESERVED unto the Lessor the several easements and other rights specified in the Third Schedule hereto TO HOLD the same unto the Lessee from the twenty fourth day of June One thousand nine hundred and ninety three for the term of twenty years thence next ensuing YIELDING AND PAYING therefore unto the Lessor during the said term hereby granted (a) during the first five years of the said term the yearly rent of Five thousand seven hundred and fifty pounds and (b) during the remainder of the said term the greater of the yearly rent of Five thousand seven hundred and fifty pounds or the yearly rent of an amount equal to and calculated and ascertained from time to time in accordance with the Fifth Schedule hereto All such rents to be paid by equal quarterly payments to be made in advance on the four usual quarter days in every year without any deduction whatsoever the first payment for the proportion of a quarter for the period from the day of to the day of One thousand nine hundred and ninety to be made on or before the signing hereof.

Table B.5 Text before restructuring

Interestingly, research has shown that we do sometimes consciously store information in propositional form—in cases when we are unable for some reason to construct a coherent mental model. One example is when information is internally contradictory (e.g. 'the chair is to the left of the bed; the bed is to the left of the table; the table is to the left of the chair'). In such circumstances the evidence suggests that we first try to construct a model (unconsciously). When we fail, we resort to remembering the information we were given. A similar effect occurs when we encounter information that is written in a confusing or unfamiliar way, such as that in Table B.5. The text is transformed in Table B.6 into a more understandable but semantically equivalent form. Note the inherently hierarchical nature of the restructured text in Figure B.6.

<table>
<tr><td rowspan="5">Definitions</td><td>The development</td><td>The property known as (address omitted), registered at the Land Registry under title number (omitted), and whose approximate boundaries are marked blue on the plan.</td></tr>
<tr><td>The shop</td><td>(address omitted), whose approximate boundaries are marked red on the plan.</td></tr>
<tr><td>The term</td><td>20 years starting on 24th June 1993.</td></tr>
<tr><td>The rent</td><td>(1) For the first 5 years: £5,750 a year; (2) Then as provided by clause 8.</td></tr>
<tr><td>Payment days</td><td>25th March, 24th June, 29th September, 25th December.</td></tr>
<tr><td>Letting</td><td colspan="2">1. The landlord lets the shop to the tenant for the term.
(Define precise boundaries, omitted from original.)
2. The tenant may:
(List extra rights from the 2nd schedule to the original.)
3. The landlord, and the tenants of the other shops in the development, may:
(List exceptions from the 3rd schedule to the original.)</td></tr>
<tr><td>Tenant's duties</td><td colspan="2">4. The tenant must pay, without deduction, in advance:
(a) on completion, if not a payment day, 1/365th of the annual rent for each day until the next payment day, and
(b) one quarter of the annual rent on each payment day.</td></tr>
</table>

Table B.6 Text after restructuring

Very often, the information we have about a situation is not contradictory but ambiguous. A key finding is that, when information is indeterminate, we still go ahead and construct a mental model (Aitkenhead and Slack 1985). The model can represent only a small range of possible interpretations of the situation, and yet we choose a particular model to work with. It appears that people have difficulty in thinking in a more abstract way (i.e. keeping part of the model 'open' until more information is received). We make and revise models all the time, so there is no lasting danger in opting for only one representation. But our models are often wrong. Johnson-Laird (1993) quotes the example of a thermostat used to control household central heating. The correct way to view a thermostat is as a feedback control unit, that adjusts the rate of heating according to the current temperature. But many people hold an inadequate view of the thermostat—as a valve. This mistake leads them to turn their central heating on and off more frequently than they need to!

An analogical mental model of a situation can be thought of as a concrete example of the situation rather than an abstract description. Johnson-Laird (1994) suggests we view a mental model as an 'instantiated schema', one possible instance of a situation that a semantic network could describe. According to this view, analogical models are 'true' mental models whilst semantic networks, categorisation hierarchies and so on merely describe the emergent relationships between the concepts that are apparently, but not actually, contained in memory (Aitkenhead and Slack 1985).

B.5 Group psychology

The study of the behaviour of individuals in small groups is known as *group dynamics* (Brown 1988). This topic has been the subject of research over many years because of its deep importance to society; we inevitably operate within groups whether at home, at work or participating in social activities. The success or failure of enterprises depends to a large extent on the effective functioning of groups, in the form of committees, project teams, departments, conferences and meetings. The behaviour of individuals is influenced strongly by the groups they belong to, and the characteristics exhibited by groups are often quite distinct from those of the groups' members. In some respects, a group can be said to operate as if it has a 'mind' of its own. It is therefore important for us to look closely at factors that determine the nature and effectiveness of group interaction, especially in areas crucial to business such as solving problems and decision-making.

Effect of task type

Research indicates that the performance of groups in problem-solving and decision-making tasks can, under the right circumstances, exceed that of the individual (Zander 1989). But under other circumstances, group performance can fall below the level of individual performance. Steiner (1972) classified tasks to help explain this relationship, as shown in Table B.7.

It is often assumed that the decisions and solutions generated by groups must be superior to those made by individuals acting alone, since 'two heads are better than one'. However, experimental evidence indicates that groups perform badly in situations where there is no clear right answer—in other words, when careful and sustained thought is required. Individuals are better able to deal with problems of this type, without the distraction that occurs during group discussion. When a complex problem must be dealt with by a group, it is suggested that the problem be split into separate parts that the group can make independent decisions about (Zander 1989). Groups fare better when the problem is one of selecting a single best solution, since the process of discussion tends to 'weed out' inferior solutions and there is an increased probability that one of the group's members will suggest the right answer. Steiner's classification of task types illustrates how the ability of a group to carry out a task depends on the type of task being tackled. According to Steiner's classification, the selection of a single solution from several alternatives is a *disjunctive* task, since the ability of the group to solve the problem is dependent on the ability of the most competent member of the group. As might be expected, research has shown that group performance can be improved overall through the training of individual group members. Experience is also important; the most effective groups are ones that have become used to working together over a long period, and groups of this type consistently outperform ad hoc groups created for specific problems.

Type of task	**Example**	**Productivity depends on**
Disjunctive tasks	Solving a puzzle problem	Most competent member (since group must decide on one of the solutions)
Conjunctive tasks	Mountain-climbing with whole team attached to the same rope	Least competent member
Additive tasks	Pushing a car stuck in the mud	Sum of individual efforts
Discretionary tasks	Estimating the number of balls in a jar	Group effort, combined as group decides (e.g. averaging, adding)

Table B.7 Types of task

One factor that has consistently been found to have a negative impact on the quality of group work is the stress caused by an undue sense of urgency. If pressure is placed on a group to produce answers quickly, then normal functioning of the group will be prevented and full consideration will not be given to all relevant issues. Research has shown that when under the anxiety created by stress, decision-makers' behaviour tends to be less flexible and imaginative than it might be otherwise be. The obvious solution to this is to avoid stress if possible, and to plan ahead so that if crisis arises, suitable plans are already in hand.

The overall performance of a group does not depend solely on the type of task undertaken and the skills of group members; it also depends heavily on the mix of attitudes in the group. *Cohesiveness*, the extent to which a group's members feel and act as if they are part of a team, is important. Below, we look in more detail at some of the factors that contribute to the effectiveness of group work, including the group members' level of participation in discussion and the kinds of thinking behind group decisions.

Factors affecting level of participation

One of the most immediate determinants of success in group problem-solving or decision-making is the willingness of the group's members to participate in discussion. Even in well-functioning groups, most of the talking is done by only about 30% of the group members (Zander 1989). The fact that proceedings tend to be dominated by a small number of individuals is not necessarily a problem, since the vocal few may, in some respects, speak for the silent majority. On the other hand, useful ideas and insights may well be lost through the reluctance of certain individuals to contribute. *Social loafing* is the general reduction in effort that tends to accompany increase in group size; as a group becomes larger, the effort expended by each member tends to decrease (Latané et al 1979). A similar phenomenon known as *cognitive loafing* has also been described (Weldon and Gargano 1988). These effects are thought to be due to the anonymity and consequent reduction in accountability that follow from increased numbers. If each individual group member believes that his or her contributions can be identified then the effects of cognitive and social loafing reduce or even disappear. Paradoxically, research on problem-solving has shown that anonymity—the same condition that can lead to social or cognitive loafing—can also lead to improved creativity, for tasks that explicitly require members to be inventive (Bartis et al 1988).

A group member may feel inhibited if the group includes individuals of a higher status, especially if those individuals are in a position of authority. Research has shown that the higher-status members in a group tend to talk more than lower-status members, but that they also tend to believe that lower-status members talk too much (Zander 1989). Domination of a group by high-status individuals may be reduced simply by discussing the issue openly with the individuals concerned. Alternatively, groups can be split into smaller groups to reduce the disparity in status levels. The use of *nominal groups*, in which members are present but do not actually interact in a face-to-face manner, can also reduce inhibition (Delbeq et al 1975).

Any approach that introduces some measure of anonymity can help to reduce inhibition. In the well-known Delphi technique, contributions are submitted anonymously and pooled for further review by the group. In the Japanese *ringi* procedure, documents are circulated for amendment repeatedly until consensus is reached (Rohlen 1975). Computer technology may be used as the vehicle for discussion, replacing face-to-face contact. Participation in computer-based conferences is much more democratic than in normal face-to-face conferences, with a far higher percentage of group members making useful contributions (Kiesler et al 1984). The nature of the contributions is also different, since group members feel less reserved and are able to make more radical or candid proposals and comments.

The way in which a group session is conducted can have a strong influence on the willingness of members to participate in discussion. The type of questions raised, and the way they are posed, can be very significant. Open questions that cannot be answered simply by stating fact or with a simple 'yes' or 'no' are more likely to stimulate group members to venture opinions. Topics for discussion should be of direct personal relevance to the group members to obtain maximum input. However, questions must not be threatening—for instance, they should not require participants to show themselves in a bad light (Zander 1989). Often, considerable care must be taken to frame questions correctly so that they meet these requirements.

The issue of personal relevance is important. Group members will be stimulated to venture opinions and offer information only if the topic being discussed is sufficiently close to their own experience or area of expertise. It is up to the discussion leader to introduce topics that are relevant. Participants can be encouraged to contribute ideas on particular topics if they have been asked to prepare thoughts beforehand and to bring them to the meeting. Similarly, exposure to relevant prior work, done either by the same group or by other groups, can be useful in helping discussants to formulate their ideas and to express them well (Zander 1989).

Having obtained contributions from reluctant group members, it is important that the desired behaviour be reinforced positively. The most powerful way to do this is to take suggestions seriously and to give them full consideration. The group leader can make a point of rewarding participants for their contributions, where appropriate, and can ensure that the ideas they express are fully aired. This should involve a comparison of the various suggestions, bringing out their differences and highlighting likely outcomes. The process by which potentially unfavourable consequences are considered has been termed *anticipatory regret* (Janis and Mann 1977) and is a necessary part of responsible decision-making.

One technique that has for many years been used to encourage participation and to help generation of ideas in a group setting is *brainstorming* (Avery and Baker 1990). In brainstorming, criticism of suggestions is prohibited and as many ideas as possible are sought. Free thinking is encouraged, and ideas are recorded for later discussion without prior judgement. Brainstorming has been shown to improve the participation level of group members, presumably because the rules forbidding criticism increase members' feelings of safety. However, it does not necessarily create better ideas, and brainstorming by individuals has been found to produce ideas of better quality and quantity than those generated in brainstorming by groups (Lamm and Trommsdorf 1973).

Polarisation

An important factor governing the quality of work done by groups is whether or not opinions converge during the decision-making process. If group members disagree strongly, and their ideas do not become reconciled, then there is a danger that no decision will be made, or that an arbitrary choice will be taken in the absence of consensus. *Polarisation* occurs when certain individuals' ideas become fixated and reinforced as discussion proceeds (Myers and Lamm 1976). Rather than listening to contrary views and modifying their thinking accordingly, these individuals take everything they hear as further evidence of the correctness of their own position. Responsibility rests with the group leader to counter this tendency by ensuring that each member gives serious consideration to alternative arguments and is willing to shift their own position if necessary.

Having a multiplicity of ideas in a discussion is usually a good thing, provided that people's ideas are not completely unchangeable. One way of preventing fixation of ideas is by requiring group members explicitly to use each others' suggestions in order to see where they lead the discussion. Polarisation is also less likely when the overall aim of discussion is explicitly to help the participants to learn or to develop their ideas rather than reaching a decision. In less goal-oriented discussions members are unlikely to feel the need to defend their own ideas, or to take entrenched positions, because each person is free to take or reject any of the ideas presented during the interaction (Burnstein and Vinokur 1975).

When a definitive decision must be reached by a group, however, it is often necessary to select a single option from a set of alternatives. In these situations disagreement can be unavoidable, and group leaders are advised to resist the temptation to seek a unanimous decision. Zander (1989) reports that a study of hypothetical juries found that the need for a unanimous decision often prevented groups from reaching any decision at all. Where possible, it is advisable to be prepared to accept a majority decision in lieu of a unanimous one.

Groupthink

In certain situations, the converse problem occurs, with apparent unanimity being reached despite inadequate consideration of alternative solutions. This phenomenon is known as *groupthink*, and is characterised by extremely cohesive groups, often with a single forceful or charismatic leader (Janis 1972). When groupthink occurs, the groups concerned may examine only a few alternatives, fail to consider possible adverse consequences, drop apparently unsatisfactory alternatives too quickly, and fail to seek expert advice or to create contingency plans in case of failure. Groupthink has been blamed for some notable political mistakes including the infamous Bay of Pigs (US) and Suez (UK) episodes. The apparent unanimity within the group is illusory because there is unexpressed uncertainty, but group members fail to raise their doubts, partly through fear of being rejected. Intelligent people convince themselves that their fears are unfounded and not worth expressing. The main antidote to groupthink is to introduce outside opinion and to encourage specifically dissenting views. Members may be asked to reconsider their decisions after 'sleeping on it'.

B.6 Conclusions

This appendix has focused on evidence from psychological research in the areas of memory, perception and comprehension, mental models, and group dynamics. The theories and findings outlined here are put to use in this research as the basis for a number of psychological principles of conceptual modelling. The principles are themselves used to justify and explain a conceptual modelling technique that is tested in the experimental part of this research. A discussion of how the psychological ideas are used to formulate principles of conceptual modelling is beyond the scope of this appendix, and the reader is referred to Chapter 3, which explores their implications for the users of conceptual modelling techniques and tools.

Appendix C
Interview Notes

The following notes were taken in the interview with modeller J (referred to in the notes as 'JA'), who conducted a series of group modelling sessions using both modelling techniques (method 'X' and method 'Y') to produce a total of eight completed models. The interview focused on the ways in which modeller J used the two modelling techniques, and on his experiences in modelling sessions when using them (see Chapter 4 for an account of the experimental method). An analysis of his comments is given in Chapter 5, and they are interpreted in the light of the other experimental evidence in Chapter 6. The interview was not transcribed verbatim but, for the most part, the words are modeller J's own.

1. JA used special 'aim' components to hold terms of reference of modelling groups (e.g. Security Group) – to give some structure to modelling sessions. These also acted as a checklist to help him ensure that he'd covered all aspects.
2. JA went to the users with models containing only aims plus some components expected to be useful. Some were generic, not specifically to do with the organisation (e.g. address, person, organisation) but others were more specific (e.g. algorithm, which is specific to the Security Group).
3. JA put notes in each model giving a description of the model and stating when it was done.
4. The Fraud Forum (group) validated the Security Group model. Version 2 of the Security Group model was produced after having spoken to the Fraud Forum people.
5. After the first few sessions JA began to put a lot of effort into preparation – looking at the group's terms of reference, web site, annual report, etc. – so as to get as many concepts as possible and to speed things up.
6. JA made good use of the graphics library and also added images to the library – some that were useful in generic way (e.g. signature, design, network point etc.)
7. Towards the end of the study JA started to colour code component windows – not green for people, red for organisations – but using different colours and colourful backgrounds to wake

people up (*"a visual jab in the ribs"*). JA felt this made it easier for people to find things visually – the colours acted as an aide-mémoire.

8. JA preferred using method 'X' tool over the traditional paper-based approach. Mentally he found method 'X' more taxing (*"in respect of knowing where you are"*) but preferred it because it seemed to be easier for the participants. He felt that maybe this was because of *"the simplified view – the graphic view was more accessible than a diagrammatic view that the participants were probably not familiar with"*. The participants were therefore more likely to accept the model (*"intuitively they are more in touch with the picture - they are not faced with a big wiring diagram, they are digesting it in chunks"*).
9. Using method 'X' JA had to call regular breaks to review the model because of not being able to see the whole model at the same time (*"if you have a method 'Y' model on the board you can quickly see where the weak areas are, and where the relationships are. But with method 'X' you tend to follow a line from component to component, and not really know if you are missing another line somewhere else"*). Therefore the models needed constant review – the participants went for coffee while JA went through the model. JA felt this carried certain risks but is not necessarily a bad thing since *"there is nothing wrong with calling regular breaks"*. However, he felt it would be difficult for a novice modeller to work out if there were areas in a model that were weak and needed more work – if the business were unfamiliar.
10. Because of long experience with traditional modelling methods, JA missed having his customary instinctive grasp of a model's status (*"you would know from a method 'Y' diagram – from experience – if it doesn't look right"*).
11. JA really would have liked to have a scribe – someone to operate the tool. JA was concerned that he *"was a bit dead"* – standing still and being quiet while he used the tool.
12. JA had initial reservations regarding using a computer-based modelling tool. His concerns mainly centred around not being able to put up flipchart papers on walls – model diagrams and definitions. Part of the idea of putting things on walls is the sense of achievement – people are surrounded by what they have produced. In the method 'X' sessions JA had to achieve the same effect by frequently reviewing what had been achieved. He felt this achieved the same thing but in a different way.
13. JA felt that he achieved more with method 'X' in the modelling sessions than he would have done using method 'Y'. But he also had the feeling that the method 'X' models needed more work after each session.
14. With both methods 'X' and 'Y', JA was able to think ahead. The fact that the method 'X' tool lets the modeller browse around when adding was useful because JA could see how a given

concept had been classified in the past. However, JA generally didn't need to look for what had been done before because he already knew.

15. Overall, JA felt that the quality of the models produced with method 'X' was better, because of improved understanding by the users. JA found the idea of classifying things (as person, organisation, document, ...) aided the participants' understanding, got them discussing things and helped them think about what each concept actually was (or could be). *"It almost doesn't matter what categories you have to choose from – it just helps to have categories so you have to have a discussion"*. As a result, the descriptions in the method 'X' models were seen more accurate and attributes were better thought out (*"with method 'X' you are asking a more specific question – "what type is ...?" – so you can get a more specific answer – not just general agreement to a description"*).
16. On request JA confirmed that he had not been exposed to literature or discussion about method 'X' or its theoretical background.
17. JA aims to define each concept as soon as it is identified (*"you can get so far down blind alleys where different people have different interpretations of a box. If you can't identify something you don't know what it is"*).
18. JA explains the modelling process to the participants using the analogy of a stores manager who has to know what he needs to store in a warehouse, how he needs to keep it, who to deliver it to, how much it costs to get it, how long it takes to get it, who has it, legal implications, etc. JA avoids stressing the computer-related aspects.
19. JA initiates modelling by asking the participants to talk about their own jobs – the processes and functions they perform (*"that's how you start the discussion: by saying "I'm going to write down the nouns in what you say"*). JA then asks relevant follow-up questions. For example, in response to Security Group Aim 1 ('lawful interception requirements') JA would ask *"interception of what?"*, and then, *"what's one of those?"*
20. JA asks about processes but finds it relatively easy to represent the responses in an object model (this is true of both methods 'X' and 'Y') (*"I try to talk to them about data in terms of processes because that's how they think about things – they think about what they do. What I find time and time again is that at the end they say 'that was really interesting, it's given me a new perspective on my job' – they are not used to thinking about their job in terms of the data. They are used to people letting them get away with vague terms and fuzzy thinking – you have to pin them down and ask ..."*).
21. After making the participants look at their jobs from a data perspective, JA then tries to get them to be very clear about the relationships between things (e.g. a document normally has one author – but can a document ever have more than one author?). Normally people think about the non-

exceptional situations and tend to answer questions wrongly based on this thinking. 'Real life' can get a bit lost in a data model – what is general or run-of the-mill is not distinguished from what is exceptional. You have to cater for the exceptional – which devalues the model itself (e.g. if a document has more than one author only twice in five years!). But it would be dangerous not to cater for the exceptions - someone has to ask the question what happens when you have more than author – what is the scale of the impact? Anything that is more input into the design process is useful information.

22. JA used the interpretation as his basic tool for looking at the overall picture. He tended to use it at the early stages for building a model – asking the participants whether each part was correct. But people quickly tired of it so he dropped it. After each workshop JA used the interpretation to correct the model. He found it essential (*"the best check"*) because of only being able to see one window at a time.

The following points refer to specific usability issues, faults or enhancements requested by modeller J to the modelling tool.

23. Sometimes the model became confusing because of the way aliases were used to describe relationships. For example, GSM Entity, To GSM Entity and From GSM Entity became three separate components (see Interface in Security Group model) whereas they are really all the same thing (this stems from a bug in the modelling tool).
24. The only slow part of the tool was the image selector – for that reason JA tended not to use the image selector during the modelling sessions. This was a shame, JA felt, since it would be a useful in-built form of mental relaxation for the modelling session participants.
25. JA used an increased font size to make the projected image more readable – the best size was about 24 point. But this meant getting to the room early and rearranging icons – because increasing the font size causes captions to overlap.
26. For some captions you have to use Windows font settings - but JA couldn't seem to get the window heading size big enough. As a result some people thought the first component in the window was the name of the window! Similar problems with enlarging button text, properties window text, etc.
27. It would be nice if the tool recorded dates on models automatically.
28. Showing relationships both ways – would be nice if the implied relationships could be visible (i.e. if the tool automatically inserted reverse relationships). This would make deductions more obvious – they are a useful labour-saving device but JA did not always know what the deduction was. JA was undecided on the question of whether the implied relationships should become real relationships or not. JA pointed out that in a workshop you don't necessarily get it right first time – you might add components and know that you will sort things out later on. So the modelling

tool should not make deductions concrete since this would create work in changing them. Instead it should just show 'ghost' relationships.

29. The tool does not make it immediately obvious whether a component is decomposed or not (or whether it is decomposable). JA took to not changing the icon for components of type data so that they would be recognisable. JA feels it would be nice to be able to distinguish them somehow – e.g. by showing them ghosted, smaller, at the bottom of the screen or by using shortcut arrows (as in Windows shortcuts) for business components.
30. JA would have liked the default font for descriptions and notes to be the font set in the 'Options -> Font' setting (but not to resize existing notes). The description popup would need to be affected by the font command and resize itself to match.
31. JA would like space for the component description on the component properties form (with the same contents as the yellow popup description). Alternatively he would like a button that lets you put in the description.
32. JA felt shortcut keys would be useful for commonly-performed actions (e.g. creating a person component).

Appendix D
Forms Used During Experiment

The following forms were completed by group modelling session participants (see Chapter 4 for experimental method). They were then subjected to qualitative and numerical analysis (Chapter 5) and interpretation alongside other experimental results (Chapter 6). The *modelling session form* was completed after each modelling session by the modelling session leader. It gives administrative details about the modelling session (required for analysis of models) and also allows the session leader to record comments and observations about the modelling technique and the session participants. *Modelling team questionnaire 1* was completed prior to the experiment by all group members (i.e. business end users taking part in group modelling sessions). It captured some useful details about the group members' backgrounds, training, etc. *Modelling team questionnaire 2* was completed by the same group members after the experiment and allowed them to express views about the process they had been through, including their impressions of the relative success or failure of the exercise.

<table>
<tr><td colspan="6">Modelling session form</td></tr>
<tr><td colspan="6">Model name: Date:</td></tr>
<tr><td colspan="6">Session no.: Start time: End time:</td></tr>
<tr><td colspan="6">Session leader:</td></tr>
<tr><td>Participant</td><td>Type of input</td><td>Level of input</td><td>Difficulties</td><td>Areas of clarity</td><td>Comments</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="6">Modelling method (circle one):
Object modelling / Method X</td></tr>
<tr><td colspan="6">Name & location of model:</td></tr>
<tr><td colspan="6">Number & description of attached diagrams/printouts:</td></tr>
<tr><td colspan="6">Comments</td></tr>
</table>

Modelling team questionnaire 1		
Name:	**Date:**	
Organisation:		
Education:		
Relevant areas of business expertise:		
Have you ever participated in an information modelling exercise? Y / N **Do you have knowledge of any information modelling techniques? Y / N** **If the answer to either of the above questions is yes, please give details**		
Use of computers (tick one or more): (circle one) **( ) Rarely/never use**	**( ) Use sometimes**	**( ) Use every day**
Have you ever written a computer program or created a database? Y / N **If so, please give brief details**		

Modelling team questionnaire 2				
Name:		**Date:**		
Organisation:				
In your view what was the purpose of the information modelling sessions?				
How valuable do you think the information modelling sessions have been? (circle one)				
1 = A waste of time	**2 = Not very valuable**	**3 = Reasonably valuable**	**4 = Quite valuable**	**5 = Very valuable**
Why?				
How well do you understand the modelling technique? (circle one)				
1 = Very poorly	**2 = Not well**	**3 = Reasonably well**	**4 = Well**	**5 = Very well**
How well do you understand the modelling software (if any)? (circle one)				
1 = Very poorly	**2 = Not well**	**3 = Reasonably well**	**4 = Well**	**5 = Very well**
How complete is the model now? (circle one)				
1 = Very incomplete	**2 = Rather incomplete**	**3 = Reasonably complete**	**4 = Nearly complete**	**5 = 100% complete**
How correct is the model now? (circle one)				
1 = Very incorrect	**2 = Rather incorrect**	**3 = Reasonably correct**	**4 = Nearly correct**	**5 = 100% correct**
What overall impressions or comments do you have of the modelling sessions?				

Appendix E
Models

This appendix includes copies of all nineteen models that were produced during the experimental part of the research, each in a late stage of development. These models were produced by expert and inexperienced modellers using both method 'X' and method 'Y' (see Chapter 4 for an outline of the experimental method). The models were then subjected to a comprehensive analysis (Chapter 5 presents the results of this analysis in tabular and graphical form) and interpreted in the light of other experimental data (Chapter 6). The method 'Y' models are included as drawn, whether produced by hand or using a CASE tool. Method 'X' models are attached in interpreted form (since there is no convenient way of expressing them on paper as they appear when using the method 'X' tool). A small number of terms have been abbreviated to reduce the length of interpreted versions where necessary. The presence of (*) in the interpretation text indicates that a relationship has been deduced by the modelling tool.

E.1 Model 1 (College Administration)

A Staff member is a person

Each Staff member has a Name
Each Staff member has a Date of Birth
Each Staff member has an Address
Each Staff member may have a Phone
Each Staff member may have a Mobile Phone

A Course is an activity

Each Course has a Course Name
Each Course has a Course content
Each Course has a Start date
Each Course has an End date
Each Course has Trainers
Each Course may have Bookings (*)

A Trainer is a Staff member

Each Trainer has a Course (*)

An Assistant is a Staff member

A Course location is a place

Each Course location has an Address
Each Course location has Rooms

A Room is a place

Each Room has a Course location
Each Room has a Facilities
Each Room has a Notes

A Client is a person

Each Client has a Name
Each Client has a Date of Birth
Each Client has an Address
Each Client may have Bookings
Each Client may have an Other information
Each Client may have Enquiries
Each Client may have a Comment
Each Client has a Phone
Each Client may have a Mobile Phone

A Booking is an activity

Each Booking has a Course
Each Booking has a Client (*)
Each Booking has an Amount Payable

An Enquiry is an activity

Each Enquiry has a Client
Each Enquiry has an Enquiry date
Each Enquiry has an Info. Sent
Each Enquiry may have Follow up calls
Each Enquiry may have a Comment

A Follow up call is an activity

Each Follow up call has an Enquiry (*)
Each Follow up call has a Comment
Each Follow up call has a Follow up date

A Name is a text value

Each Name has a Surname
Each Name has a First Name

A Surname is a text value

A First Name is a text value

A Date of Birth is a date/time

A Course Name is a text value

A Course content is a text value

A Start date is a date/time

An End date is a date/time

An Address is a text value

Each Address has an Address 1
Each Address has an Address 2
Each Address has an Address 3
Each Address has a City
Each Address has a Country
Each Address has a Postcode

An Address 1 is a text value

An Address 2 is a text value

An Address 3 is a text value

A City is a text value

A Country is a text value

A Postcode is a text value

A Facilities is a text value

A Notes is a text value

A Money Amount is a number

A Rate is a number

A Cost is a number

An Other information is a text value

An Amount Payable is a number

An Enquiry date is a date/time

An Info. Sent is a date/time

A Comment is a text value

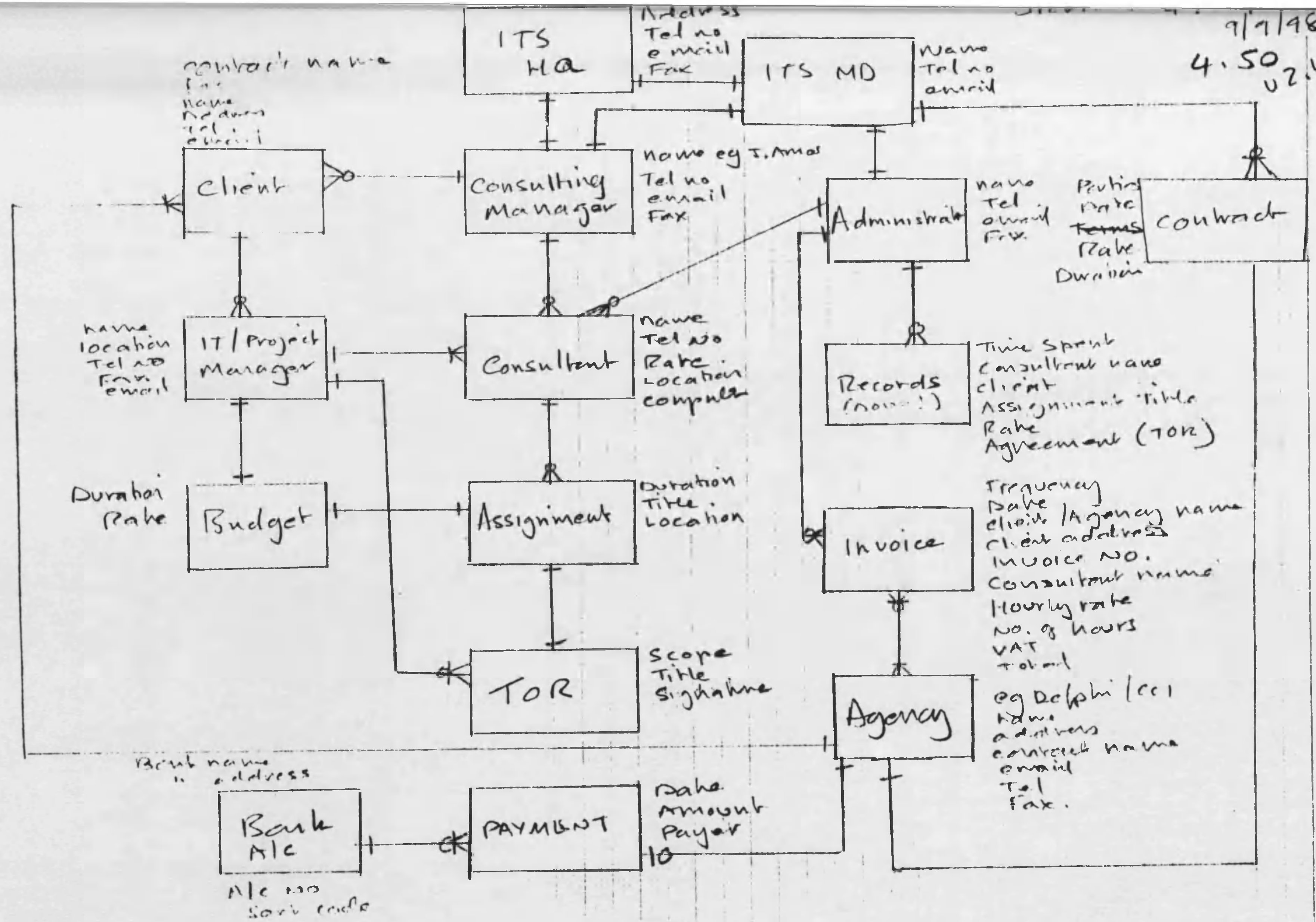
A Follow up date is a date/time

A Phone is a number

A Mobile Phone is a number

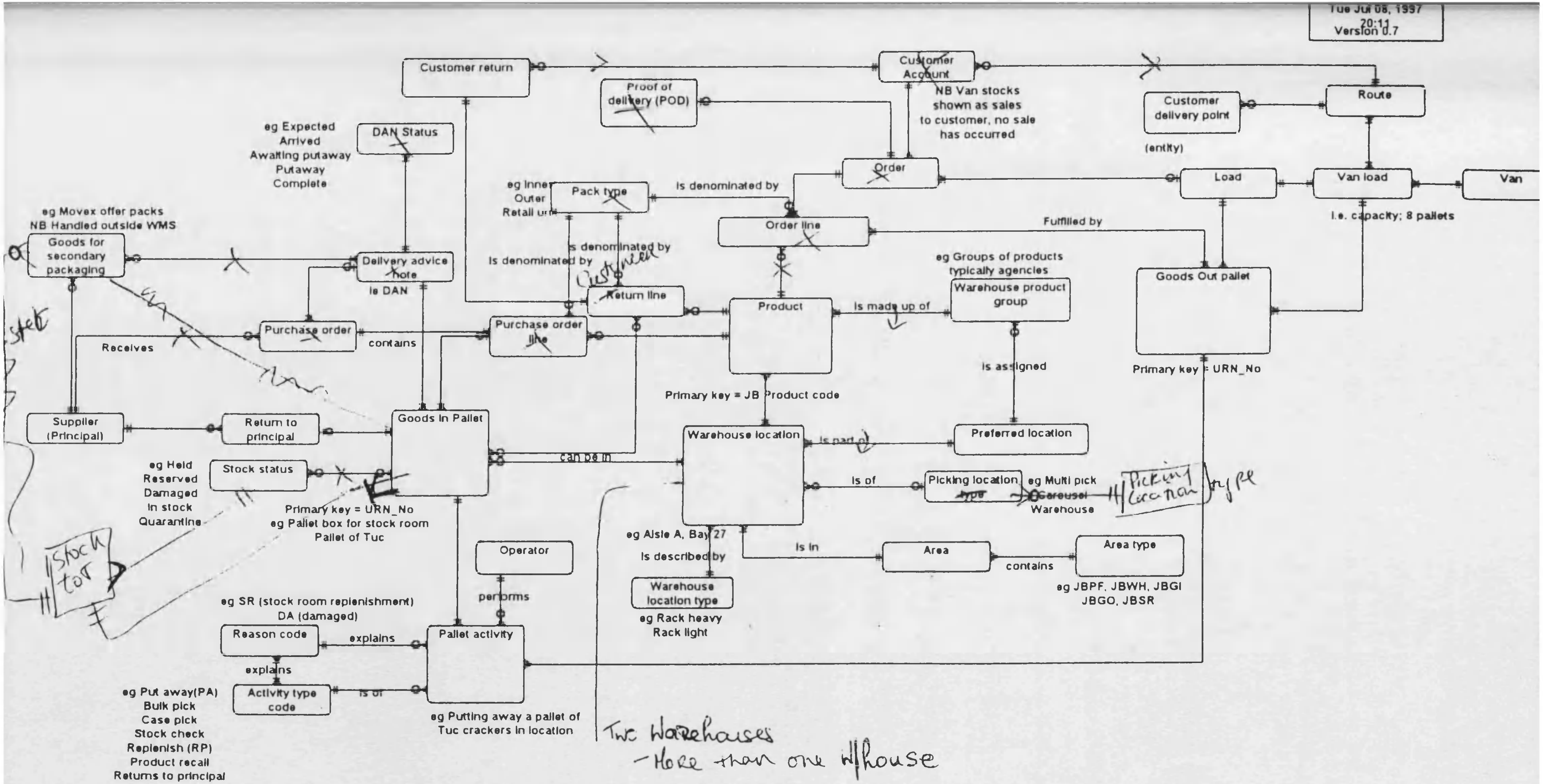
E.2 Model 2 (Consulting)

9/9/98
4.50
v2.4



ITS HQ
Address
Tel no
e mail
Fax
ITS MD
Name
Tel no
email
contract name
Client
Consulting Manager
name eg J. Amos
Tel no
email
Fax
Administrator
name
Tel
email
Fax
Parties
Date
~~Terms~~
Rate
Duration
Contract
name
location
Tel no
Fax
email
IT / Project Manager
Consultant
name
Tel no
Rate
Location
company
Records (none!)
Time Spent
Consultant name
client
Assignment Title
Rate
Agreement (TOR)
Duration
Rate
Budget
Assignment
Duration
Title
Location
Invoice
Frequency
Date
client / Agency name
client address
Invoice NO.
Consultant name
Hourly rate
No. of hours
VAT
Total
TOR
Scope
Title
Signature
Agency
eg Delphi / CCI
name
address
contract name
email
Tel
Fax.
Bank name
" address
Bank A/c
A/c no
Sort code
PAYMENT
Date
Amount
Payer

E.3 Model 3 (Distribution Warehousing)


Tue Jul 08, 1997 20:11
Version 0.7
Customer return
Customer Account
Route
Proof of delivery (POD)
NB Van stocks shown as sales to customer, no sale has occurred
Customer delivery point
(entity)
eg Expected
Arrived
Awaiting putaway
Putaway
Complete
DAN Status
Order
Load
Van load
Van
eg Inner
Outer
Retail unit
Pack type
is denominated by
i.e. capacity; 8 pallets
eg Movex offer packs
NB Handled outside WMS
Order line
Fulfilled by
Goods for secondary packaging
Delivery advice note
is denominated by
is denominated by
Customer
eg Groups of products typically agencies
is DAN
Return line
Goods Out pallet
Product
is made up of
Warehouse product group
Receives
Purchase order
contains
Purchase order line
is assigned
Primary key = URN_No
Primary key = JB Product code
Supplier (Principal)
Return to principal
Goods In Pallet
Warehouse location
is part of
Preferred location
can be in
eg Held
Reserved
Damaged
In stock
Quarantine
Stock status
is of
Picking location type
eg Multi pick
Carousel
Warehouse
Picking location type
Primary key = URN_No
eg Pallet box for stock room
Pallet of Tuc
eg Aisle A, Bay 27
is in
Area
Area type
Operator
is described by
contains
Stock tor
Warehouse location type
eg JBPF, JBWH, JBGI
JBGO, JBSR
eg SR (stock room replenishment)
DA (damaged)
performs
eg Rack heavy
Rack light
Reason code
explains
Pallet activity
explains
Activity type code
is of
eg Put away(PA)
Bulk pick
Case pick
Stock check
Replenish (RP)
Product recall
Returns to principal
eg Putting away a pallet of Tuc crackers in location
Two Warehouses
- More than one W/house

E.4 Model 4 (Fraud Forum)

A Network is an object

A Security Algorithm is an activity

A Protocol is an activity

An Equipment Item is an object

A Law is a document

A Country is a place

A Region is a place

A Fraud Type is a category

An Equipment Type is a category

A Supplier is an organisation

A Counter Measure is a concept

An Audit is an activity

An Accreditation Scheme is a concept

A Scheme Status is a category

An Aim is a concept

An Aim1 is a concept

Each Aim1 has a Description

An Aim2 is a concept

Each Aim2 has a Description2

An Aim3 is a concept

Each Aim3 has a Description3

An Aim4 is a concept

Each Aim4 has a Description4

An Aim5 is a concept

Each Aim5 has a Description5

An Aim6 is a concept

Each Aim6 has a Description6

An Aim7 is a concept

Each Aim7 has a Description7

An Aim8 is a concept

Each Aim8 has a Description8

An Aim9 is a concept

Each Aim9 has a Description9

An Aim10 is a concept

Each Aim10 has a Description10

An Aim11 is a concept

Each Aim11 has a Description11

An Aim12 is a concept

Each Aim12 has a Description12

A Description is a notes

Each Description may have Aim1s (*)

A Description2 is a notes

Each Description2 may have Aim2s (*)

A Description3 is a notes

Each Description3 may have Aim3s (*)

A Description4 is a notes

Each Description4 may have Aim4s (*)

A Description5 is a notes

Each Description5 may have Aim5s (*)

A Description6 is a notes

Each Description6 may have Aim6s (*)

A Description7 is a notes

Each Description7 may have Aim7s (*)

A Description8 is a notes

Each Description8 may have Aim8s (*)

A Description9 is a notes

Each Description9 may have Aim9s (*)

A Description10 is a notes

Each Description10 may have Aim10s (*)

A Description11 is a notes

Each Description11 may have Aim11s (*)

A Description12 is a notes

Each Description12 may have Aim12s (*)

A README is a notes

E.5 Model 5 (Fund Management)

An Investor is a person
Each Investor may have a Company location
Each Investor has Investment instructions
Each Investor has Shareholder letters (*)
Each Investor has Dividend payments (*)
Each Investor has an Investment Objective
Each Investor has Unit issues (*)
Each Investor has a Name
Each Investor has an Investor type
Each Investor has a Processes

An Investment instruction is a document
Each Investment instruction has an Investor
Each Investment instruction has an Amount invested
Each Investment instruction has a Fund
Each Investment instruction may have a Deal
Each Investment instruction has a Date

A Fund is an organisation
Each Fund has a Company location
Each Fund has Investment instructions (*)
Each Fund has a Shareholder Services Co.
Each Fund has a Global Custodian Bank
Each Fund has a Fund Accounting Company
Each Fund has a Fund name
Each Fund has Portfolio holdings
Each Fund has Deals (*)
Each Fund has a Board of Directors
Each Fund has a Custody System
Each Fund has an Accounting system
Each Fund has Shareholder letters (*)
Each Fund has Risks
Each Fund has Annual reports
Each Fund has Dividends
Each Fund has an Investment Management Co.
Each Fund has Unit issues
Each Fund has Fund bank accounts (*)
Each Fund has a Record number
Each Fund has Miscellaneous payments
Each Fund has a Minimum investment amount
Each Fund has a Fund type

A Shareholder Services Co. is an organisation
Each Shareholder Services Co. has a Company location
Each Shareholder Services Co. has Funds (*)
Each Shareholder Services Co. has Employees
Each Shareholder Services Co. has Shareholder letters
Each Shareholder Services Co. has a Company name
Each Shareholder Services Co. has Investment advices

A Global Custodian Bank is an organisation
Each Global Custodian Bank has a Company location
Each Global Custodian Bank has Funds (*)
Each Global Custodian Bank has Employees
Each Global Custodian Bank has Fund bank accounts
Each Global Custodian Bank has a Company name
Each Global Custodian Bank has Employee payments
Each Global Custodian Bank has a Depository Bank
Each Global Custodian Bank has a Sub-Custodian

A Fund Accounting Company is an organisation
Each Fund Accounting Company has a Location
Each Fund Accounting Company has Funds (*)
Each Fund Accounting Company has Employees
Each Fund Accounting Company has a Company name

A Portfolio holding is a document
Each Portfolio holding has a Fund (*)
Each Portfolio holding has a Security
Each Portfolio holding has a Date
Each Portfolio holding has a Brokerage Firm (*)
Each Portfolio holding has a Quantity held

An Employee is a person
Each Employee may have a Shareholder Services Co.(*)
Each Employee may have a Global Custodian Bank (*)
Each Employee may have a Fund Accounting Co. (*)
Each Employee may have a Board of Directors (*)
Each Employee may have an Investment Mgt Co. (*)
Each Employee has a Name
Each Employee has a Telephone number
Each Employee has an Address
Each Employee may have a Brokerage Firm (*)
Each Employee has Employee payments (*)
Each Employee has a Job title
Each Employee may have a Depository Bank (*)

A Deal is an activity
Each Deal has Investment instructions (*)
Each Deal has a Fund
Each Deal has a Settlement date
Each Deal has a Security
Each Deal has a Trade details document
Each Deal has a Trade date
Each Deal may have an Investment Management Co. (*)
Each Deal has a Brokerage Firm
Each Deal has an Amount
Each Deal has Broker payments (*)

A Security is a document
Each Security has a Company location
Each Security has Portfolio holdings (*)
Each Security has Deals (*)
Each Security has a Net Asset Value
Each Security has a Security name
Each Security has Price quotes (*)
Each Security has a Security type text
Each Security has a Security type
Each Security has Share certificates (*)
Each Security has Dividend receipts

A Trade details document is a document
Each Trade details document has a Deal
Each Trade details document has a File ref.

A Board of Directors is an organisation
Each Board of Directors has a Company location
Each Board of Directors has Funds (*)
Each Board of Directors has Employees
Each Board of Directors has a Company name

A Custody System is a system
Each Custody System has Funds (*)
Each Custody System has a Corporate action text
Each Custody System has a System name
Each Custody System has a Systems text
Each Custody System has a System activity

An Accounting system is a system
Each Accounting system has Funds (*)
Each Accounting system has a System name
Each Accounting system has a Systems text
Each Accounting system has a System activity

A Shareholder letter is a document
Each Shareholder letter has an Investor
Each Shareholder letter has a Fund
Each Shareholder letter has a Shareholder Serv. Co. (*)
Each Shareholder letter has a Letter

A Letter is a document
Each Letter has Shareholder letters (*)
Each Letter has a Date
Each Letter has a Letter text

A Dividend payment is an activity
Each Dividend payment has an Investor
Each Dividend payment has a Dividend amount

Each Dividend payment has a Dividend (*)
Each Dividend payment has a Payable date

A Risk is a concept
Each Risk has a Fund (*)
Each Risk has a Risk type
Each Risk has a Name of risk
Each Risk has a Description

A Risk type is a notes
Each Risk type has Risks (*)

A Fund location is a place
Each Fund location has a Street
Each Fund location has a Country
Each Fund location has a Telephone number
Each Fund location has a Fax number
Each Fund location has a City
Each Fund location has a House number

An Annual report is a document
Each Annual report has a Fund (*)
Each Annual report has a Date
Each Annual report has an Annual report name
Each Annual report has an Economic overview desc.
Each Annual report has an Annual profit/loss

A Dividend is an activity
Each Dividend has a Fund (*)
Each Dividend has Dividend payments
Each Dividend has a Declaration date
Each Dividend has a Dividend amount
Each Dividend has a Dividends

An Investment Management Co. is an organisation
Each Investment Management Co. has a Location
Each Investment Management Co. has Funds (*)
Each Investment Management Co. has Employees
Each Investment Management Co. has Deals
Each Investment Management Co. has a Name
Each Investment Management Co. has Share certificates

A Fund & regulatory restriction is a document
Each Fund & regulatory restriction has a Description
Each Fund & regulatory restriction has a Dep. Bank (*)

An Investment Objective is a notes
Each Investment Objective has Investors (*)

A Corporate action text is a notes
Each Corporate action text has Custody Systems (*)

A Unit issue is an activity
Each Unit issue has an Investor
Each Unit issue may have a Fund (*)
Each Unit issue has a Date
Each Unit issue has a Number of units

A Fund bank acc. is a document
Each Fund bank acc. has a Fund
Each Fund bank acc. may have a Global Cust. Bank (*)
Each Fund bank acc. has an Account Number
Each Fund bank acc. has an Account type
Each Fund bank acc. has a Cash in bank

A Price quote is an activity
Each Price quote has a Security
Each Price quote has a Date
Each Price quote has a Brokerage Firm (*)
Each Price quote has a Price

A Fund type examples is a notes
Each Fund type examples has Fund types (*)

A Brokerage Firm is an organisation
Each Brokerage Firm has a Company location
Each Brokerage Firm has Portfolio holdings
Each Brokerage Firm has Employees
Each Brokerage Firm has Deals (*)
Each Brokerage Firm has Price quotes
Each Brokerage Firm has a Company name
Each Brokerage Firm has Broker payments

A Security type text is a notes
Each Security type text has Securities (*)

An Expenses is a notes

A Dividends is a notes
Each Dividends has Dividends (*)

An Account type is a document
Each Account type has Fund bank accounts (*)
Each Account type has an Account name

An Employee payment is an activity
Each Employee payment has a Company location
Each Employee payment has a Global Cust. Bank (*)
Each Employee payment has an Employee
Each Employee payment has a Payment amount
Each Employee payment has a Date
Each Employee payment has a Company name

A Security type is a document
Each Security type has Securities (*)
Each Security type has a Description

An Economic overview desc. is a document
Each Economic overview desc. has Annual reports (*)
Each Economic overview desc. has a Description

A Systems text is a notes
Each Systems text has Custody Systems (*)
Each Systems text has Accounting systems (*)

A Share cert. is a document
Each Share cert. has a Security
Each Share cert. has a Date
Each Share cert. may have an Investment Mgt Co. (*)

A Depository Bank is an organisation
Each Depository Bank has a Company location
Each Depository Bank has Global Custodian Banks (*)
Each Depository Bank has Employees
Each Depository Bank has Fund & regulatory restrictions
Each Depository Bank has a Company name

A Sub-Custodian is a notes
Each Sub-Custodian has Global Custodian Banks (*)

A Dividend receipt is a document
Each Dividend receipt has a Security (*)
Each Dividend receipt has a Date
Each Dividend receipt has a Dividend amount
Each Dividend receipt may have an Interest receivable

A Processes is a doc. link
Each Processes has Investors (*)

A Glossary is a doc. link

A Fee glossary is a notes
Each Fee glossary has Miscellaneous payments (*)

A Payee is a person
Each Payee has a Company location
Each Payee has a Name
Each Payee has a Company name
Each Payee has a Job title
Each Payee has Miscellaneous payments (*)

A Miscellaneous payment is an activity
  Each Miscellaneous payment has a Fund (*)
  Each Miscellaneous payment has a Payment amount
  Each Miscellaneous payment has a Date
  Each Miscellaneous payment has a Description
  Each Miscellaneous payment has a Fee glossary
  Each Miscellaneous payment has a Payee
  Each Miscellaneous payment has a Payment name

A Fund type is a category
  Each Fund type has Funds (*)
  Each Fund type has a Fund type examples
  Each Fund type has a Description

A Broker payment is an activity
  Each Broker payment has a Deal
  Each Broker payment has a Payment amount
  Each Broker payment has a Date
  Each Broker payment has a Brokerage Firm

A System activity is a notes
  Each System activity has Custody Systems (*)
  Each System activity has Accounting systems (*)

A Letter text is a document
  Each Letter text has Letters (*)
  Each Letter text has a Net Asset Value
  Each Letter text has a Description
  Each Letter text has a Letter ref. number

An Investment advice is a document
  Each Investment advice has a Shareholder Serv. Co. (*)
  Each Investment advice has a Description

A Company location is a text value
  Each Company location has a Telephone number
  Each Company location has a Fax number
  Each Company location has an Address

An Amount invested is a text value
A Fund name is a text value
A Settlement date is a date/time
A Number of securities purchased is a number
An Account Number is a number
A First name is a text value
A Surname is a text value
A Payment amount is a text value
A Net Asset Value is a text value
A File ref. is a text value
A Date is a date/time
A Declaration date is a date/time
A Trade date is a date/time
A Dividend amount is a text value
A Reinvestment date is a date/time
A Name of risk is a text value
A Security name is a text value
An Annual report name is a text value
A Cost of security is a number
A Total value of fund is a number
A Street is a text value
A Country is a text value
A Location name is a text value
A Description is a text value
A Name is a text value
  Each Name has a First name
  Each Name has a Surname
  Each Name has a Title
A Title is a text value
A Telephone number is a text value
A Fax number is a text value
A City is a text value
A Record number is a number
An Address is a text value
  Each Address has a Street
  Each Address has a Country
  Each Address has a City
  Each Address has a House number
A House number is a number
A Number of units is a number
A Security location is a text value
A Company name is a text value
An Investor type is a text value
A Price is a number
A Distribution date is a date/time
A Ref. number is a number
A Cash in bank is a text value
An Interest receivable is a number
A Job title is a text value
A System name is a text value
An Amount is a number
A Quantity held is a number
A Payable date is a date/time
A Record date is a date/time
An Annual profit/loss is a text value
A Meeting date is a date/time
A Minimum investment amount is a text value
A Letter ref. number is a number
An Account name is a text value
A Payment name is a text value

E.6 Model 6 (Help Desk)

A User is a person

Each User has a Name
Each User has an Emp ID
Each User may have a Tel No.
Each User has a Manager
Each User may have Problem reports

A Manager is a person

Each Manager has Users
Each Manager has a Name
Each Manager has a Tel No.

A Problem report is a document

Each Problem report has an User
Each Problem report has a Rpt #
Each Problem report may have a Technician
Each Problem report has a Received
Each Problem report may have a Completed
Each Problem report may have a Description
Each Problem report may have a Resolution
Each Problem report has a Status
Each Problem report may have a Mgr Signature
Each Problem report may have a Priority
Each Problem report has a SubType
Each Problem report has a Type (*)
Each Problem report may have an Amount
Each Problem report may have an IT Signature

A Technician is a person

Each Technician has a Name
Each Technician has Problem reports (*)
Each Technician may have Skills
Each Technician has an Escalator (*)

A Skill is a category

Each Skill has a Technician (*)
Each Skill has a Skill Level
Each Skill has a Type

A Skill Level is a category

Each Skill Level may have Skills (*)
Each Skill Level has a Level Text

A SubType is a category

Each SubType has a Name
Each SubType may have Problem reports
Each SubType has a SLA
Each SubType may have a Type

A Type is a category

Each Type has a Name
Each Type may have Problem reports
Each Type may have Skills (*)
Each Type may have SubTypes

An Escalator is a person

Each Escalator has a Name
Each Escalator may have Technicians

A Name is a text value

An Emp ID is a number

A Tel No. is a text value

A Problem is a text value

A Rpt # is a number

A New Text is a text value

A Received is a date/time

A Completed is a date/time

An Elapsed is a number

A SLA is a number

A Description is a text value

A Resolution is a text value

A Status is a text value

A Skill Text is a text value

A Level Text is a text value

A Mgr Signature is a text value

A Priority is a number

An Amount is a number

An IT Signature is a text value

E.7 Model 7 (Homeopathic Remedies)

A Remedy is a category

Each Remedy has a Name
Each Remedy has an Abbreviation
Each Remedy has a Source
Each Remedy may have Affinities
Each Remedy may have Aetiologies
Each Remedy may have Modalities
Each Remedy has Mentals
Each Remedy has Physical Generals
Each Remedy may have Physical Particulars
Each Remedy may have Remedy indications (*)

A Symptom is a category

Each Symptom has a Description
Each Symptom may have Sensations
Each Symptom may have Locations
Each Symptom may have Colours
Each Symptom may have Remedy indications (*)

A Symptom Category is a category

Each Symptom Category has a Description
Each Symptom Category has a Weighting
Each Symptom Category may have Rem. Ind. (*)

An Affinity is a Location

Each Affinity may have a Remedy (*)

An Aetiology is a Remedy indication

Each Aetiology may have a Remedy (*)

A Modality is a Remedy indication

Each Modality may have a Remedy (*)

A Mental is a Remedy indication

Each Mental may have a Remedy (*)

A Physical General is a Remedy indication

Each Physical General may have a Remedy (*)

A Physical Particular is a Remedy indication

Each Physical Particular may have a Remedy (*)

A Sensation is a category

Each Sensation has a Symptom (*)

A Location is a category

Each Location has a Symptom (*)

A Colour is a category

Each Colour has a Symptom (*)

A Remedy indication is a category

Each Remedy indication has a Remedy
Each Remedy indication has a Symptom
Each Remedy indication has a Symptom Cat.
Each Remedy indication has a Weighting

A Name is a text value

An Abbreviation is a text value

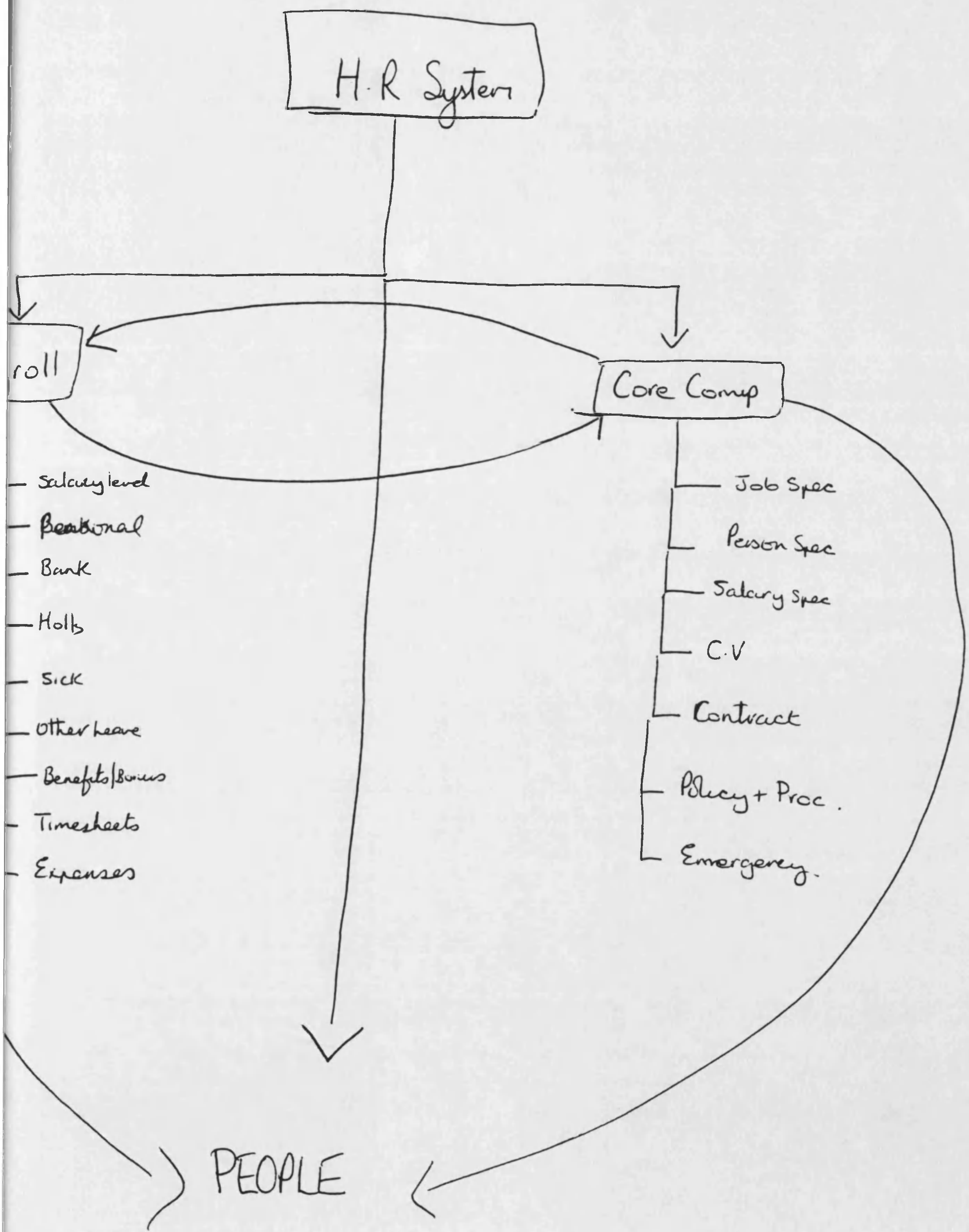
A Source is a text value

A Description is a text value

A New Text is a text value

A Weighting is a number

E.8 Model 8 (Human Resources)


H.R System
roll
Core Comp
Salary level
Bank
Holls
Sick
Other Leave
Benefits/Bonus
Timesheets
Expenses
Job Spec
Person Spec
Salary Spec
C.V
Contract
Policy + Proc.
Emergency.
PEOPLE

E.9 Model 9 (International Roaming)

An Intro is a notes

A TOR Item 1 is a concept
Each TOR Item 1 has an Item 1 Text
Each TOR Item 1 has an Operator

An Item 1 Text is a notes
Each Item 1 Text may have TOR Item 1s (*)

A TOR Item 2 is a concept
Each TOR Item 2 has an Item 2 Text
Each TOR Item 2 has a Call Case
Each TOR Item 2 has a Test Specification

An Item 2 Text is a notes
Each Item 2 Text may have TOR Item 2s (*)

A TOR Item 3 is a concept
Each TOR Item 3 has an Item 3 Text
Each TOR Item 3 has a Route

An Item 3 Text is a notes
Each Item 3 Text may have TOR Item 3s (*)

A TOR Item 4 is a concept
Each TOR Item 4 has an Item 4 Text
Each TOR Item 4 has a Service

An Item 4 Text is a notes
Each Item 4 Text may have TOR Item 4s (*)

A TOR Item 5 is a concept
Each TOR Item 5 has an Item 5 Text

An Item 5 Text is a notes
Each Item 5 Text may have TOR Item 5s (*)

An Operator is an organisation
Each Operator may have TOR Item 1s (*)
Each Operator has an Operator Definition
Each Operator has a Name
Each Operator has Networks
Each Operator has a Country
Each Operator has Operator Services

An Operator Definition is a notes
Each Operator Definition may have Operators (*)

A Network is an object
Each Network has an Operator
Each Network has Network Addresses
Each Network has Network Services
Each Network has Signalling Plans
Each Network has a Network Type
Each Network has Frequency Band Spectra
Each Network has a Coverage Area

A Network Address is a place
Each Network Address has a Network
Each Network Address has a Gateway Y/N
Each Network Address may have a Signalling Plan (*)
Each Network Address may have Exchange Equip. Items
Each Network Address has Routes

A Gateway Y/N is a category
Each Gateway Y/N has a Name
Each Gateway Y/N has Network Addresses
Each Gateway Y/N has a Description

A Service is an activity
Each Service may have TOR Item 4s (*)
Each Service has a Name
Each Service has a Service Definition
Each Service has Network Services
Each Service may have Coverage Areas
Each Service has a Service Type
Each Service has Operator Services

A Service Definition is a notes
Each Service Def may have Services (*)

A Network Service is an object
Each Network Service has a Network
Each Network Service has a Service

A Signalling Plan is a document
Each Signalling Plan may have a Network (*)
Each Signalling Plan has Network Addresses
Each Signalling Plan has a From network
Each Signalling Plan has a To Network

A From network is a Network
Each From network may have a Signalling Plan (*)

A To Network is a Network
Each To Network may have a Signalling Plan (*)

A Network Type is a category
Each Network Type has a Name
Each Network Type has Networks
Each Network Type has a Network Type Definition
Each Network Type has a Description

A Network Type Definition is a notes
Each Network Type Def may have Network Types (*)

A Frequency Band Spectrum is a concept
Each Frequency Band Spectrum may have a Network (*)
Each Frequency Band Spectrum has a Freq. Band Definition
Each Frequency Band Spectrum has a Lower Frequency
Each Frequency Band Spectrum has an Upper Frequency

A Frequency Band Definition is a notes
Each Frequency Band Def may have Freq. Band Spectra (*)

A Subscriber is a person
Each Subscriber has a Name
Each Subscriber has a Subscriber Definition
Each Subscriber has a Subscriber Number
Each Subscriber has Operator Service Subscribers

A Subscriber Definition is a notes
Each Subscriber Def may have Subscribers (*)

A Country is a place
Each Country may have Operators
Each Country has a Name

A Coverage Area is a place
Each Coverage Area has a Network
Each Coverage Area has Services
Each Coverage Area has a Coverage Area Definition
Each Coverage Area has Maps

A Coverage Area Definition is a notes
Each Coverage Area Def may have Coverage Areas (*)

A Map is a Document
Each Map may have a Coverage Area (*)

A Service Type is a category
Each Service Type has a Name
Each Service Type may have Services (*)
Each Service Type has a Description
Each Service Type has a Service Type Definition

A Service Type Definition is a notes
Each Service Type Def may have Service Types (*)

A Call Case is a concept
Each Call Case may have TOR Item 2s (*)
Each Call Case has a Name
Each Call Case has a Call Case Definition
Each Call Case has Tests
Each Call Case has a From Equipment Type
Each Call Case has a To Equipment Type

A Call Case Definition is a notes

Each Call Case Def may have Call Cases (*)

A Test Specification is a document
Each Test Specification may have TOR Item 2s (*)
Each Test Specification has a Description
Each Test Specification has a Test spec Definition
Each Test Specification has an Identifier
Each Test Specification may have Tests (*)

A Test spec Definition is a notes
Each Test spec Def may have Test Specifications (*)

A Permanent Reference Doc is a document
Each Permanent Reference Doc may have Standards (*)

A Standard is a concept
Each Standard has a Permanent Reference Doc

A Test is an activity
Each Test has a Call Case
Each Test has a Test Specification
Each Test has a Test Definition
Each Test has a From Operator
Each Test has a To Operator
Each Test has a Test Date
Each Test has Test Results
Each Test has a Tester
Each Test may have an Exchange Equipment Item
Each Test has a Person (*)

A Test Definition is a notes
Each Test Def may have Tests (*)

A From Operator is an Operator
Each From Operator may have a Test (*)

A To Operator is an Operator
Each To Operator may have a Test (*)

A Test Result is a concept
Each Test Result has a Test
Each Test Result has a Test Result Variety
Each Test Result has a Value
Each Test Result has a TADIG Co-ordinator

A Test Result Variety is a category
Each Test Result Variety may have Test Results (*)
Each Test Result Variety has a Variety Definition

A Variety Definition is a notes
Each Variety Def may have Test Result Varieties (*)

A Tester is a Person
Each Tester may have a Test (*)

A TADIG Co-ordinator is a Person
Each TADIG Co-ordinator may have a Test Result (*)

An Exchange Equipment is an object

An Exchange Equipment Item is an object
Each Exchange Equip. Item may have a Network Address
Each Exchange Equipment Item has Tests (*)
Each Exchange Equipment Item has a Supplier
Each Exchange Equipment Item has a Model
Each Exchange Equipment Item has a serial number
Each Exchange Equipment Item has an Equipment Type

A Supplier is an organisation
Each Supplier may have Exchange Equipment Items (*)

A Route is a concept
Each Route may have TOR Item 3s (*)
Each Route may have a Network Address (*)
Each Route has an Identifier
Each Route has a From Network Address
Each Route has a To Network Address
Each Route may have Intermediate Network Addresses
Each Route has a Preference Level

A From Network Address is a Network Address
Each From Network Address may have a Route (*)

A To Network Address is a Network Address
Each To Network Address may have a Route (*)

An Intermediate Network Address is a Network Address
Each Intermediate Network Address may have a Route (*)

A Person is a person
Each Person has a Name
Each Person may have Tests
Each Person may have Operator Service Subscribers (*)
Each Person has a email Address
Each Person has a Phone Number

An Equipment Type is a category
Each Equipment Type has a Name
Each Equipment Type has a Description
Each Equipment Type may have Exchange Eqpt Items (*)

A From Equipment Type is an Equipment Type
Each From Equipment Type may have a Call Case (*)

A To Equipment Type is a category
Each To Equipment Type may have Call Cases (*)

An Operator Service is a concept
Each Operator Service has an Operator
Each Operator Service has a Service
Each Operator Service has Operator Service Subscribers

An Operator Service Subscriber is a person
Each Op. Service Subscriber may have a Subscriber (*)
Each Operator Service Subscriber has a Person
Each Operator Service Subscriber has an Operator Service

A Risk is a concept
A New Notes is a notes
A Document is a document
A Name is a text value
A Lower Frequency is a number
A Upper Frequency is a number
A Description is a text value
A Subscriber Number is a number
An Identifier is a number
A Test Date is a date/time
A Value is a text value
A Preference Level is a text value
A Model is a text value
A serial number is a number
An email Address is a text value
A Phone Number is a number

E.10 Model 10 (Legal)

9.1 Providing Legal Advice is an activity

9.2 Controlling Trademark is an activity
Each 9.2 Controlling Trademark has a Trade Mark

9.3 EU Notification is an activity
Each 9.3 EU Notification has a Notification
Each 9.3 EU Notification has a Notification Definition

9.4 Resolving issues of Competition law is an activity

9.5 Conf. Agreement is an activity
Each 9.5 Conf. Agreement has an Agreement
Each 9.5 Conf. Agreement has an Application for Restricted docs
Each 9.5 Conf. Agreement has an Authoriser
Each 9.5 Conf. Agreement has an Application to attend meeting

9.6 Co-operation Agreement is an activity

9.7 Third Party Contracts is an activity

An Organisation is an organisation
Each Organisation has a Name
Each Organisation has a Company Number
Each Organisation has an Industry
Each Organisation has Addresses
Each Organisation has an Agreement
Each Organisation has License to uses (*)
Each Organisation may have Application for Use (*)
Each Organisation has Third Party Application's (*)
Each Organisation may have Application for Restricted doc (*)
Each Organisation has an Org Type
Each Organisation may have Application to attend meetings (*)

An Industry is a category
Each Industry may have Organisations (*)
Each Industry has a Name
Each Industry has a Description

An Address is a place
Each Address may have an Organisation (*)
Each Address has an Address Line1
Each Address has an Address Line 2
Each Address has an Address Line 3
Each Address has a City
Each Address has a County State
Each Address has a Country
Each Address may have a Person (*)

A Law is a document

A logo is a concept

An Agreement is a document
Each Agreement may have 9.5 Conf. Agreements (*)
Each Agreement may have Organisations (*)
Each Agreement may have Signatures (*)
Each Agreement has Authorisers

A Trade Mark is a concept
Each Trade Mark may have 9.2 Controlling Trademarks (*)
Each Trade Mark has a Name
Each Trade Mark has a Trademark Definition
Each Trade Mark has a Trademark Classification
Each Trade Mark has Condition of Uses
Each Trade Mark has a License to use
Each Trade Mark has Application for Uses
Each Trade Mark has Design Components
Each Trade Mark has Registrations

A Person is a person
Each Person has a Name
Each Person has Addresses
Each Person has a Telephone Number
Each Person has a email address
Each Person has a Gender
Each Person has an Employer
Each Person may have Signatures (*)

A Gender is a category
Each Gender may have Persons (*)

An Employer is an Organisation
Each Employer may have a Person (*)

A Signature is a document
Each Signature has an Agreement
Each Signature has a Person

A Legal & Regulatory Group is an organisation
Each Legal & Regulatory Group has a L and R TOR

A L and R TOR is a notes
Each L and R TOR may have Legal & Regulatory Groups (*)

A Trademark Definition is a notes
Each Trademark Def may have Trade Marks (*)

A Trademark Classification is a category
Each Trademark Classification has a Name
Each Trademark Classification has a Description
Each Trademark Classification may have Trade Marks (*)
Each Trademark Classification has an Example of Classification

An Example of Classification is a notes
Each Example of Classification may have Trademark Class's (*)

A Trademark Registry is an Organisation
Each Trademark Registry may have a Registration (*)

A Condition of Use is a document
Each Condition of Use has a Description
Each Condition of Use may have a Trade Mark (*)
Each Condition of Use has an Identifier

A License to use is a document
Each License to use may have an Organisation
Each License to use may have Trade Marks (*)
Each License to use has a LRO4
Each License to use has a LR05
Each License to use has a LR06
Each License to use has a Business activity
Each License to use may have a GSM Member
Each License to use has a LR02
Each License to use has a LR03

A LRO4 is a notes
Each LRO4 may have License to uses (*)

A LR05 is a notes
Each LR05 may have License to uses (*)

A LR06 is a notes
Each LR06 may have License to uses (*)

A Business activity is an activity
Each Business activity may have License to uses (*)

A GSM Member is an organisation
Each GSM Member has License to uses (*)
Each GSM Member has Third Party Application's (*)

A LR02 is a notes
Each LR02 may have License to uses (*)

A LR03 is a notes
Each LR03 may have License to uses (*)

An Application for Use is a document
Each Application for Use has an Organisation
Each Application for Use may have a Trade Mark (*)
Each Application for Use has Example Designs
Each Application for Use has an Application Date

An Example Design is a concept
Each Example Design may have an Application for Use (*)

A Design Component is a concept
Each Design Component may have a Trade Mark (*)
Each Design Component has Colour Uses
Each Design Component may have Font Uses
Each Design Component may have a Size Ratio
Each Design Component has an Identifier

A Colour Use is a concept
Each Colour Use has a Design Component
Each Colour Use has a Colour Value
Each Colour Use has a Colour

A Font Use is a concept

Each Font Use has a Design Component
Each Font Use has a Text Spacing
Each Font Use has a Colour
Each Font Use has a Size
Each Font Use has a Font

A Registration is a document
Each Registration has a Country
Each Registration may have a Trade Mark (*)
Each Registration has a Trademark Registry
Each Registration has Third Party Application's
Each Registration has an Identifier
Each Registration has a Registration Definition

A Third Party Application is a document
Each Third Party Application may have an Organisation
Each Third Party Application may have a GSM Member
Each Third Party Application may have a Registration (*)
Each Third Party Application has a Deadline
Each Third Party Application may have an Opposition
Each Third Party Application may have a Communication

An Opposition is an activity
Each Opposition has Third Party Application's (*)

A Communication is an activity
Each Communication has Third Party Application's (*)

A Document Application is a document
Each Document Application may have 9.5 Conf. Agreements (*)
Each Document Application has an Organisation
Each Document Application has a Documents

A Documents is a document
Each Documents may have Application for Restricted docs (*)

A Document Type is a category
Each Document Type has an Examples

An Examples is a notes
Each Examples may have Document Types (*)

An Org Type is a category
Each Org Type may have Organisations (*)
Each Org Type has an In type Examples

A City is a text value
A County State is a text value
A Country is a text value
A Telephone Number is a number
An email address is a text value
A Colour Value is a number
A Text Spacing is a number

An In type Examples is a notes
Each In type Examples may have Org Types (*)

An Authoriser is a person
Each Authoriser may have 9.5 Conf. Agreements (*)
Each Authoriser may have an Agreement (*)

A Submission to EC is an activity
Each Submission to EC may have Notifications (*)
Each Submission to EC has a Date

An Application to attend meeting is a document
Each Application to attend meeting may have 9.5 Conf. Agreements (*)
Each Application to attend meeting has an Organisation

A Notification is a document
Each Notification may have 9.3 EU Notifications (*)
Each Notification has a Submission to EC
Each Notification has Signatories

A Notification Definition is a notes
Each Notification Def may have 9.3 EU Notification (*)

A Signatory is a Person
Each Signatory may have a Notification (*)

A Colour is a concept
Each Colour may have Colour Uses (*)
Each Colour may have Font Uses (*)

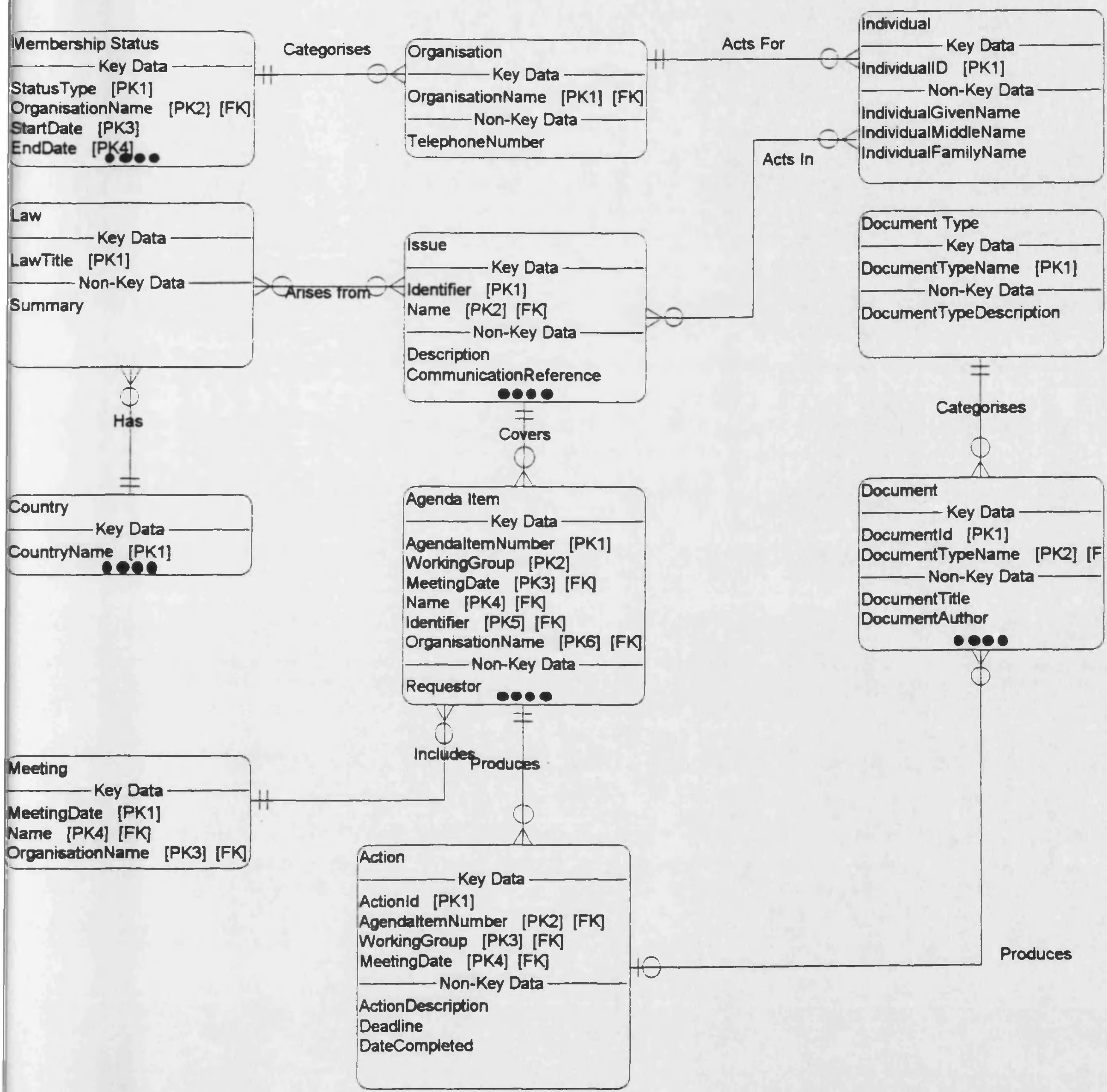
A Font is a concept
Each Font may have Font Uses (*)

A Registration Definition is a notes
Each Registration Definition may have Registrations (*)

A Trademark Registry Definition is a notes

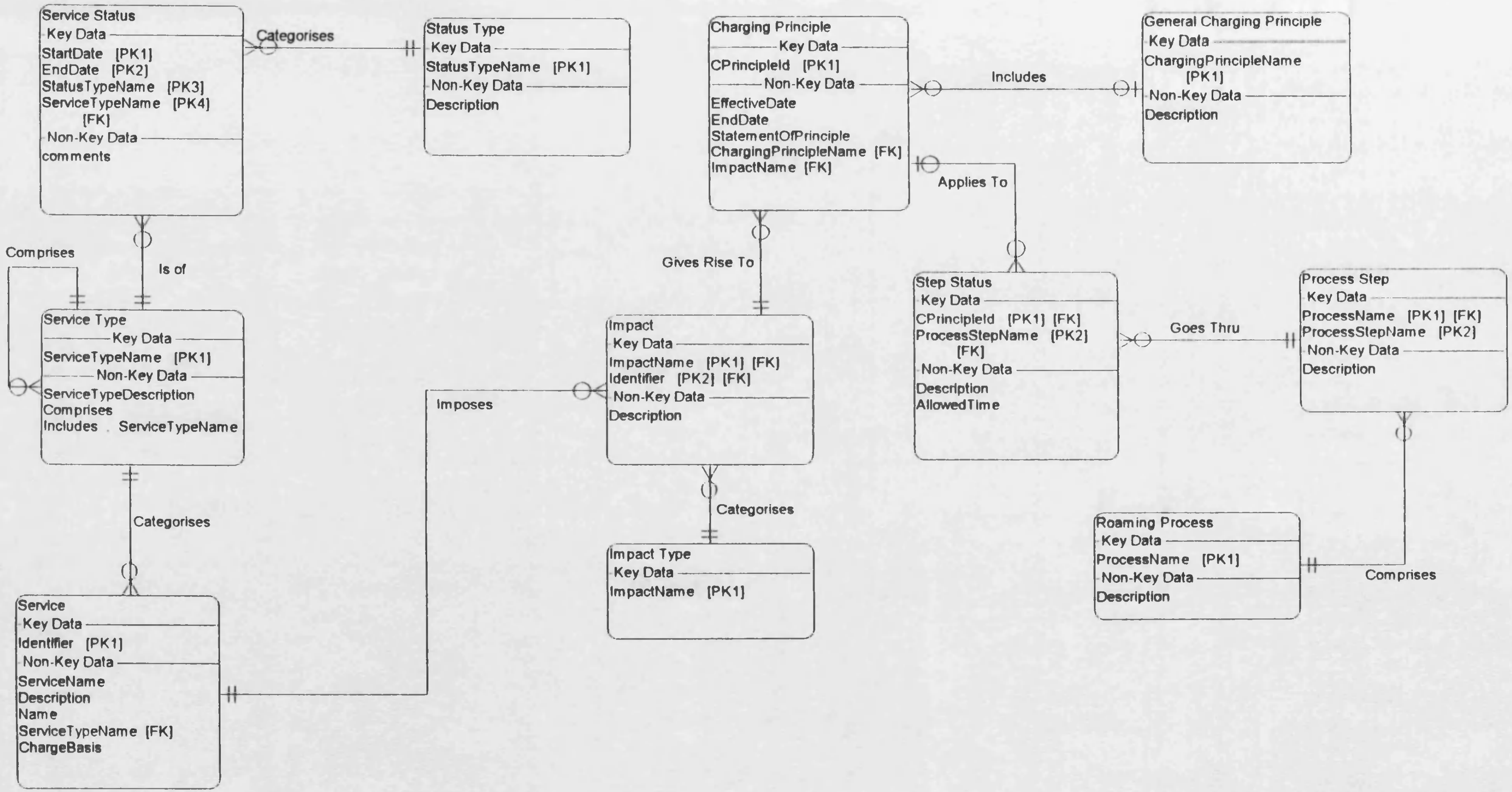
A Name is a text value
A Company Number is a number
A Description is a text value
An Address Line is a text value
An Address Line1 is a text value
An Address Line 2 is a text value
An Address Line 3 is a text value
A Size Ratio is a text value
A Deadline is a date/time
A Date is a date/time
A Size is a number
An Identifier is a number
An Application Date is a date/time

E.11 Model 11 (Legal and Regulatory)



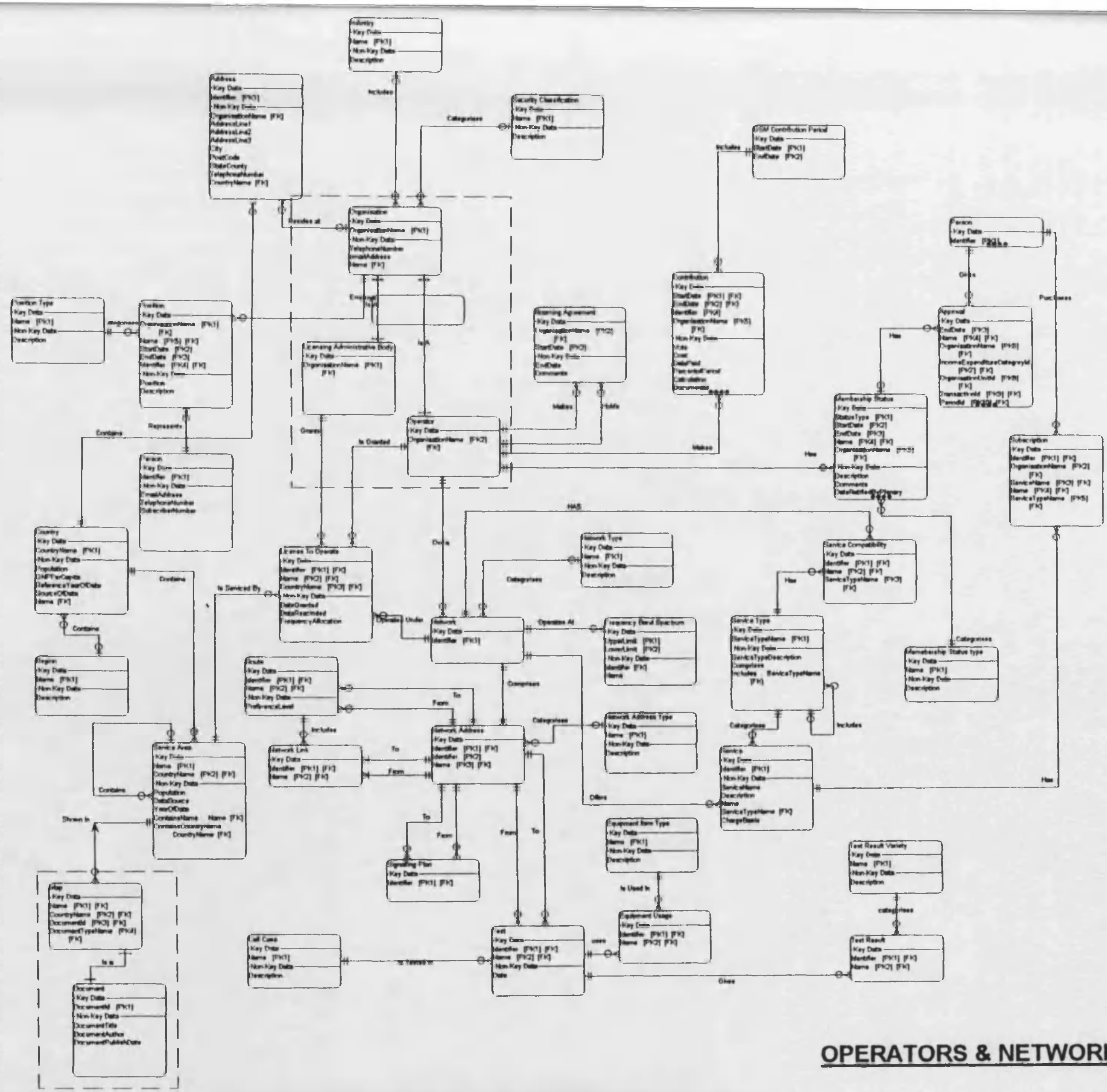
L & R - Issues & Advice

E.12 Model 12 (Mobile Phone Billing)



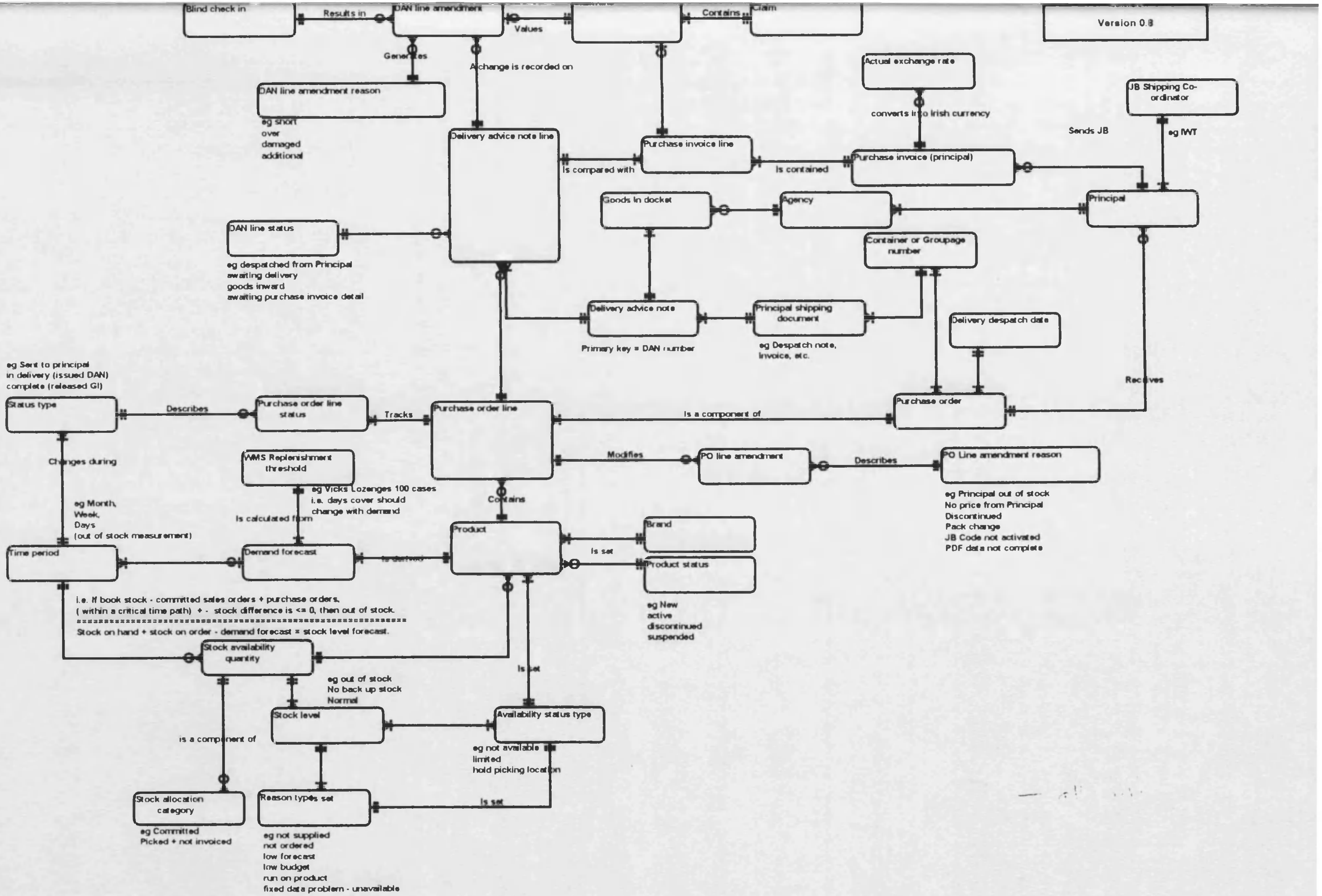
BARG - Charging Principles V 1.0

E.13 Model 13 (Operators and Networks)



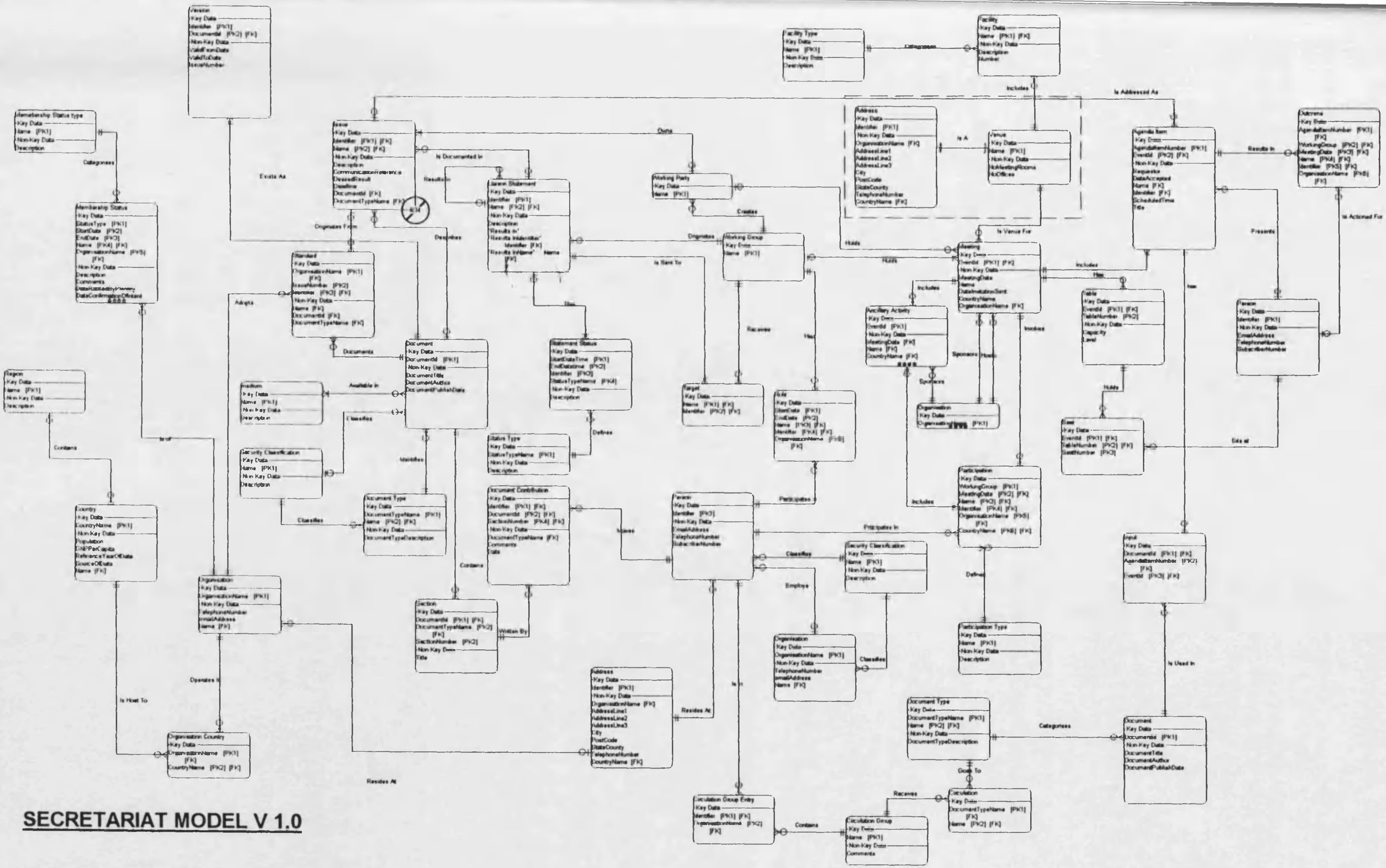
OPERATORS & NETWORKS V 1.0

E.14 Model 14 (Purchase Orders)



Blind check in
Results in
DAN line amendment
Values
Contains
Claim
Version 0.8
Generates
A change is recorded on
Actual exchange rate
JB Shipping Co-ordinator
DAN line amendment reason
eg short
over
damaged
additional
converts into Irish currency
Sends JB
eg IWT
Delivery advice note line
Purchase invoice line
Is compared with
Is contained
Purchase invoice (principal)
Goods in docket
Agency
Principal
DAN line status
eg despatched from Principal
awaiting delivery
goods inward
awaiting purchase invoice detail
Container or Groupage number
Delivery advice note
Principal shipping document
Delivery despatch date
Primary key = DAN number
eg Despatch note, invoice, etc.
Receives
eg Sent to principal
in delivery (issued DAN)
complete (released GI)
Status type
Describes
Purchase order line status
Tracks
Purchase order line
Is a component of
Purchase order
Changes during
WMS Replenishment threshold
Modifies
PO line amendment
Describes
PO Line amendment reason
eg Vicks Lozenges 100 cases
i.e. days cover should change with demand
Contains
eg Principal out of stock
No price from Principal
Discontinued
Pack change
JB Code not activated
PDF data not complete
eg Month,
Week,
Days
(out of stock measurement)
Is calculated from
Product
Brand
Is set
Time period
Demand forecast
Is derived
Product status
i.e. If book stock - committed sales orders + purchase orders,
(within a critical time path) + - stock difference is <= 0, then out of stock.
Stock on hand + stock on order - demand forecast = stock level forecast.
eg New
active
discontinued
suspended
Stock availability quantity
Is set
eg out of stock
No back up stock
Normal
Stock level
Availability status type
is a component of
eg not available
limited
hold picking location
Stock allocation category
Reason types set
Is set
eg Committed
Picked + not invoiced
eg not supplied
not ordered
low forecast
low budget
run on product
fixed data problem - unavailable

E.15 Model 15 (Secretariat)



SECRETARIAT MODEL V 1.0

E.16 Model 16 (Security Group)

An Aim1 is a concept
Each Aim1 has a Description1
Each Aim1 has a Standard
Each Aim1 has a Liaison Statement
Each Aim1 has an Operator Rep
Each Aim1 has a Product

A Description1 is a notes
Each Description1 may have Aim1s (*)

An Aim2 is a concept
Each Aim2 has a Description2
Each Aim2 has an Attack

A Description2 is a notes
Each Description2 may have Aim2s (*)

An Aim3 is a concept
Each Aim3 has a Description3
Each Aim3 has a GSM Entity
Each Aim3 has an Interface

An Aim4 is a concept
Each Aim4 has a Description4

A Description3 is a notes
Each Description3 may have Aim3s (*)

A Description4 is a notes
Each Description4 may have Aim4s (*)

An Aim5 is a concept
Each Aim5 has a Description5

A Description5 is a notes
Each Description5 may have Aim5s (*)

A Technical Protocol is an activity
Each Technical Protocol has Std. (*)
Each Technical Protocol has a Name
Each Technical Protocol has a Technical Protocol Def.
Each Technical Protocol has a Protocol Description
Each Technical Protocol has Interfaces
Each Technical Protocol has Documenting Std.

A Country is a place
Each Country may have Regions
Each Country has a Name
Each Country may have GSM MoU Members
Each Country may have Algorithm Types
Each Country has an International Dialling Code

A Region is a place
Each Region has Countries
Each Region has a Name
Each Region may have GSM MoU Members
Each Region has a Region Definition
Each Region has a Description

An Equipment Type is a category
Each Equipment Type has a Name
Each Equipment Type has an Equipment Type Definition
Each Equipment Type has a Description

A Supplier is an organisation
Each Supplier may have Audits
Each Supplier has a Telephone No.
Each Supplier has an Address
Each Supplier may have GSM Entities
Each Supplier has a Supplier Definition
Each Supplier may have Products

A Counter Measure is a concept
Each Counter Measure may have an Attack (*)
Each Counter Measure has a Counter Measure Type
Each Counter Measure has Threats
Each Counter Measure has an Implementation Cost
Each Counter Measure has a Failure Cost
Each Counter Measure has a Failure Likelihood %
Each Counter Measure has an Annual Loss Expectancy
Each Counter Measure may have an Algorithm
Each Counter Measure has a Counter Measure Definition
Each Counter Measure may have a Review Date
Each Counter Measure may have Product Measures (*)
Each Counter Measure has Assoc. Technical Protocols
Each Counter Measure may have Impl. Prod. Measures
Each Counter Measure has a Description

An Audit is an activity
Each Audit has a Supplier
Each Audit has a Person (*)
Each Audit has Processes (*)
Each Audit has a Date & Time
Each Audit has a Publish Date
Each Audit may have a Review Date
Each Audit may have Products
Each Audit has an Audit Definition
Each Audit may have Business Processes
Each Audit may have Premises
Each Audit may have Employees
Each Audit may have IT Systems
Each Audit has an Auditor
Each Audit may have a Certificate
Each Audit has an Audit Rep
Each Audit has a Company (*)

A Standard is a concept
Each Standard may have Aim1s (*)
Each Standard may have a Technical Protocol
Each Standard has an Identifier
Each Standard has a Standard Definition
Each Standard has Versions
Each Standard has a Std. Body
Each Standard may have an Original Issue
Each Standard has an Orig. Counter Measure
Each Standard has a Current Version
Each Standard has Products

An Attack is an activity
Each Attack may have Aim2s (*)
Each Attack has Counter Measures
Each Attack may have Issues
Each Attack has an Attack Definition
Each Attack has a Method Description
Each Attack has Threats

A Person is a person
Each Person may have Audits
Each Person has a Name
Each Person has a email address
Each Person has a Telephone No.
Each Person has an Address
Each Person has a Working Group
Each Person has a GSM MoU Member
Each Person may have Meeting's
Each Person may have Agenda Items
Each Person may have an Outcome
Each Person has an Access Type

An Address is a place
Each Address may have Suppliers (*)
Each Address may have a Person
Each Address has an Address Line 1
Each Address has an Address Line 2
Each Address has an Address Line 3
Each Address has a City
Each Address has a Postal Code
Each Address may have Std. Bodies (*)

Each Address may have GSM MoU Members (*)
Each Address may have Meeting's (*)
Each Address may have Sources (*)
Each Address may have a Company

A Liaison Statement is an activity
Each Liaison Statement may have Aim1s (*)
Each Liaison Statement has a Liaison Statement Def.
Each Liaison Statement may have Issues
Each Liaison Statement has an Identifier
Each Liaison Statement has a Statement Status
Each Liaison Statement has Dest. Working Groups
Each Liaison Statement has an Orig. Working Group
Each Liaison Statement may have Resulting Liaison Stats

A Liaison Statement Definition is a notes
Each Liaison Statement Def may have Liaison Stmts. (*)

An Issue is a concept
Each Issue may have an Attack
Each Issue has a Name
Each Issue may have a Liaison Statement
Each Issue has an Issue Definition
Each Issue has an Issue Description
Each Issue may have an Input Document
Each Issue may have a Working Party (*)
Each Issue has a Desired Result
Each Issue has a Deadline
Each Issue may have an Owning Working Party

A Working Group is an organisation
Each Working Group may have Persons (*)
Each Working Group has a Name
Each Working Group has a Chairman
Each Working Group has Working Group Meeting's
Each Working Group has Working Parties
Each Working Group may have Meeting's (*)
Each Working Group has Delegates

A Statement Status is a category
Each Statement Status may have Liaison Statements. (*)
Each Statement Status has a Description
Each Statement Status has a Status Type
Each Statement Status has a Start Date Time
Each Statement Status has an End Date Time

An Issue Definition is a notes
Each Issue Def may have Issues (*)

A Standard Definition is a notes
Each Standard Def may have Std. (*)

A Version is a concept
Each Version has a Standard
Each Version has an Identifier
Each Version has a Publish Date
Each Version has Authors

A Std. Body is an organisation
Each Std. Body has Std.
Each Std. Body has a email address
Each Std. Body has a Telephone No.
Each Std. Body has an Address
Each Std. Body has a Std. Body Definition

An Attack Definition is a notes
Each Attack Def may have Attacks (*)

A Process is an activity
Each Process may have an Audit
Each Process has a Name
Each Process has a Process Definition
Each Process has a Process Type
Each Process has a Process Description
Each Process may have Threats

A Process Definition is a notes
Each Process Def may have Processes (*)

A Process Type is a category
Each Process Type has a Name
Each Process Type may have Processes (*)
Each Process Type has a Description

A Counter Measure Type is a category
Each Counter Measure Type has Counter Measures
Each Counter Measure Type has a Name
Each Counter Measure Type has a Description

A Threat is a concept
Each Threat has Counter Measures
Each Threat has an Attack
Each Threat has a Process
Each Threat has a PR Impact
Each Threat has a Legal Impact
Each Threat has a Financial Impact

An Algorithm is an activity
Each Algorithm has Counter Measures
Each Algorithm has a Name
Each Algorithm has an Identifier
Each Algorithm has an Algorithm Definition
Each Algorithm has GSM MoU Members
Each Algorithm has a PRD
Each Algorithm has an Export License
Each Algorithm has an Algorithm Type
Each Algorithm has a Publish Date
Each Algorithm may have a Fee

An Algorithm Definition is a notes
Each Algorithm Def may have Algorithms (*)

A GSM MoU Member is an organisation
Each GSM MoU Member has Countries
Each GSM MoU Member has a Region
Each GSM MoU Member has Persons
Each GSM MoU Member has a Name
Each GSM MoU Member has an Address
Each GSM MoU Member has Algorithms
Each GSM MoU Member may have Export Licenses
Each GSM MoU Member has a Member Definition
Each GSM MoU Member has Operator Reps

A PRD is a document
Each PRD has an Identifier
Each PRD may have an Algorithm

An Export License is a document
Each Export License has an Identifier
Each Export License may have Algorithms
Each Export License has GSM MoU Members
Each Export License has an Expiry Date

An Algorithm Type is a category
Each Algorithm Type may have Countries
Each Algorithm Type has a Name
Each Algorithm Type has Algorithms
Each Algorithm Type has an Algorithm Type Description

A Technical Protocol Definition is a notes
Each Technical Protocol Def may have Technical Protocols (*)

An Equipment Type Definition is a notes
Each Equipment Type Def may have Equipment Types (*)

A GSM Entity is an object
Each GSM Entity may have Aim3s (*)
Each GSM Entity has Suppliers
Each GSM Entity has a Name

Each GSM Entity has a GSM Entity Definition
Each GSM Entity has a GSM Entity Description
Each GSM Entity has Interfaces

A GSM Entity Definition is a notes
Each GSM Entity Def may have GSM Entities (*)

A Supplier Definition is a notes
Each Supplier Def may have Suppliers (*)

An Interface is a concept
Each Interface may have Aim3s (*)
Each Interface has Technical Protocols
Each Interface may have a GSM Entity (*)
Each Interface has a From GSM Entity
Each Interface has a To GSM Entity
Each Interface has an Interface Definition

A Model Description is a notes

A Dest. Working Group is a Working Group
Each Dest. Working Group may have a Liaison Stmt. (*)

An Orig. Working Group is a Working Group
Each Orig. Working Group may have a Liaison Stmt. (*)

A Resulting Liaison Stmt. is a Liaison Statement
Each Resulting Liaison Stmt.may have a Liaison Stmt. (*)

An Author is a Person
Each Author may have a Version (*)

A Std. Body Definition is a notes
Each Std. Body Def may have Std. Bodies (*)

An Original Issue is an Issue
Each Original Issue may have a Standard (*)

An Orig. Counter Measure is a Counter Measure
Each Orig. Counter Measure may have a Standard (*)

A Counter Measure Definition is a notes
Each Counter Measure Def may have Counter Meas. (*)

A From GSM Entity is a GSM Entity
Each From GSM Entity may have an Interface (*)

A To GSM Entity is a GSM Entity
Each To GSM Entity may have an Interface (*)

A Chairman is a Person
Each Chairman may have a Working Group (*)

An Equipment is an object
Each Equipment has a Name

A Current Version is a Version
Each Current Version may have a Standard (*)

A Working Group Meeting is a Meeting
Each Working Group Meeting may have a Wkg. Group (*)

An Agenda is a document

An Input Document is a document
Each Input Document has Issues
Each Input Document has Agenda Items
Each Input Document has Sources
Each Input Document has a Title
Each Input Document has a Document Reference

A Working Party is an organisation
Each Working Party has Issues
Each Working Party has a Working Group
Each Working Party may have Meeting's (*)
Each Working Party has Working Party Meeting's

A Meeting is an activity
Each Meeting has a Person (*)
Each Meeting has an Address
Each Meeting has a Working Group
Each Meeting has a Date & Time
Each Meeting has a Working Party
Each Meeting has Agenda Items
Each Meeting has an Attendee

A Working Party Meeting is a Meeting
Each Working Party Meeting may have a Wkg. Party (*)

An Agenda Item is an activity
Each Agenda Item has a Person (*)
Each Agenda Item has a Name
Each Agenda Item may have Input Documents
Each Agenda Item has a Meeting
Each Agenda Item may have a Presenter
Each Agenda Item may have Outcomes

A Source is a person
Each Source has a Name
Each Source has a email address
Each Source has a Telephone No.
Each Source has an Address
Each Source has Input Documents

An Attendee is a Person
Each Attendee may have a Meeting (*)

A Presenter is a Person
Each Presenter may have an Agenda Item (*)

An Outcome is an activity
Each Outcome has Persons (*)
Each Outcome has an Agenda Item
Each Outcome may have Assigned persons
Each Outcome has an Assigned Working Group
Each Outcome has an Outcome Description

An Assigned person is a Person
Each Assigned person may have an Outcome (*)

An Assigned Working Group is a Working Group
Each Assigned Working Group may have an Outcome (*)

A Delegate is a Person
Each Delegate may have a Working Group (*)

An Access Type is a category
Each Access Type has Persons
Each Access Type has a Name
Each Access Type has an Access Type Definition

An Access Type Definition is a notes
Each Access Type Def may have Access Types (*)

A Member Definition is a notes
Each Member Def may have GSM MoU Members (*)

A Region Definition is a notes
Each Region Def may have Regions (*)

An Operator Rep is a document
Each Operator Rep may have Aim1s (*)
Each Operator Rep has a GSM MoU Member
Each Operator Rep has a Publish Date
Each Operator Rep has an Operator Rep Definition
Each Operator Rep has a Description

An Operator Rep Definition is a notes
Each Operator Rep Def may have Operator Reps (*)

A Product is an object
Each Product may have Aim1s (*)
Each Product has a Supplier
Each Product has Audits
Each Product may have Std.
Each Product has a Name
Each Product has a Product Type
Each Product has Product Measures
Each Product has an Evaluation
Each Product has a Price
Each Product may have a Standard Guaranteed

A Product Type is a category
Each Product Type has a Name
Each Product Type may have Products (*)
Each Product Type has a Product Type Definition
Each Product Type has a Description

A Product Measure is a concept
Each Product Measure has a Counter Measure
Each Product Measure has a Product
Each Product Measure has an Evaluation
Each Product Measure has a Product Measure Definition

A Product Type Definition is a notes
Each Product Type Def may have Product Types (*)

An Audit Definition is a notes
Each Audit Def may have Audits (*)

A Business Process is a Process
Each Business Process may have an Audit (*)

A Premises is an Address
Each Premises may have an Audit (*)

An Employee is a Person
Each Employee may have an Audit (*)

An IT System is a system
Each IT System may have Audits
Each IT System has a Name
Each IT System has a Description

An Auditor is a Company
Each Auditor may have an Audit (*)

A Certificate is a document
Each Certificate has Audits
Each Certificate has a Certificate Number
Each Certificate has an Expiry Date

An Audit Rep is a document
Each Audit Rep has an Audit
Each Audit Rep has an Identifier

A Documenting Standard is a Standard
Each Documenting Standard may have a Technical Protocol (*)

An Interface Definition is a notes
Each Interface Def may have Interfaces (*)

An Original Attack is an Attack

An Assoc. Technical Protocol is a Technical Protocol
Each Assoc. Technical Protocol may have a Counter Measure (*)

An Impl. Product Measure is a Product Measure
Each Impl. Product Meas. may have a Counter Meas. (*)

A Company is an organisation
Each Company may have Audits
Each Company has a Name
Each Company has a Telephone No.
Each Company has an Address

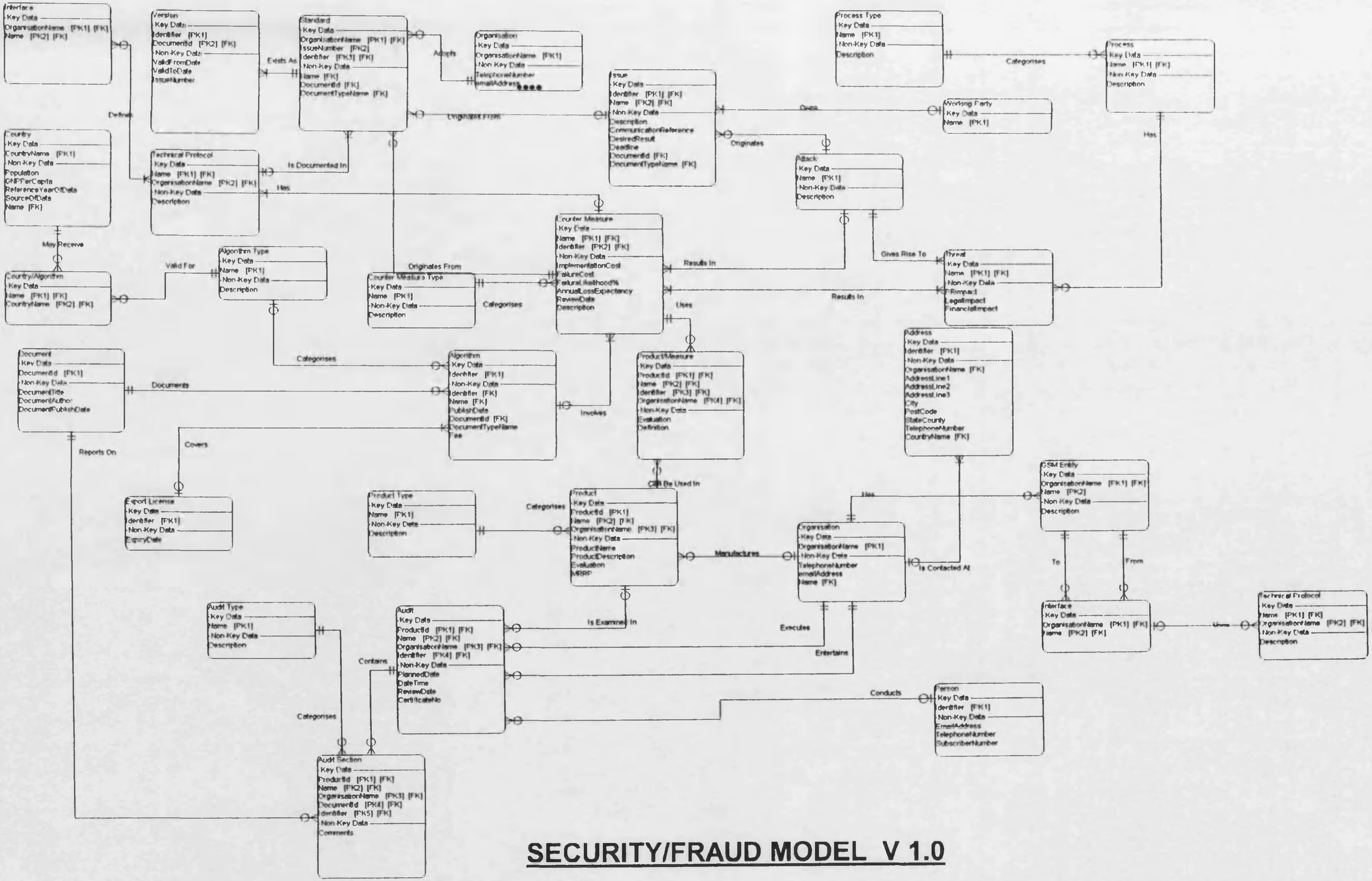
A Status Type is a category
Each Status Type has a Name
Each Status Type may have Statement Statuses (*)
Each Status Type has a Description

An Owning Working Party is a Working Party
Each Owning Working Party may have an Issue (*)

A Product Measure Definition is a notes
Each Product Measure Def may have Product Meas. (*)

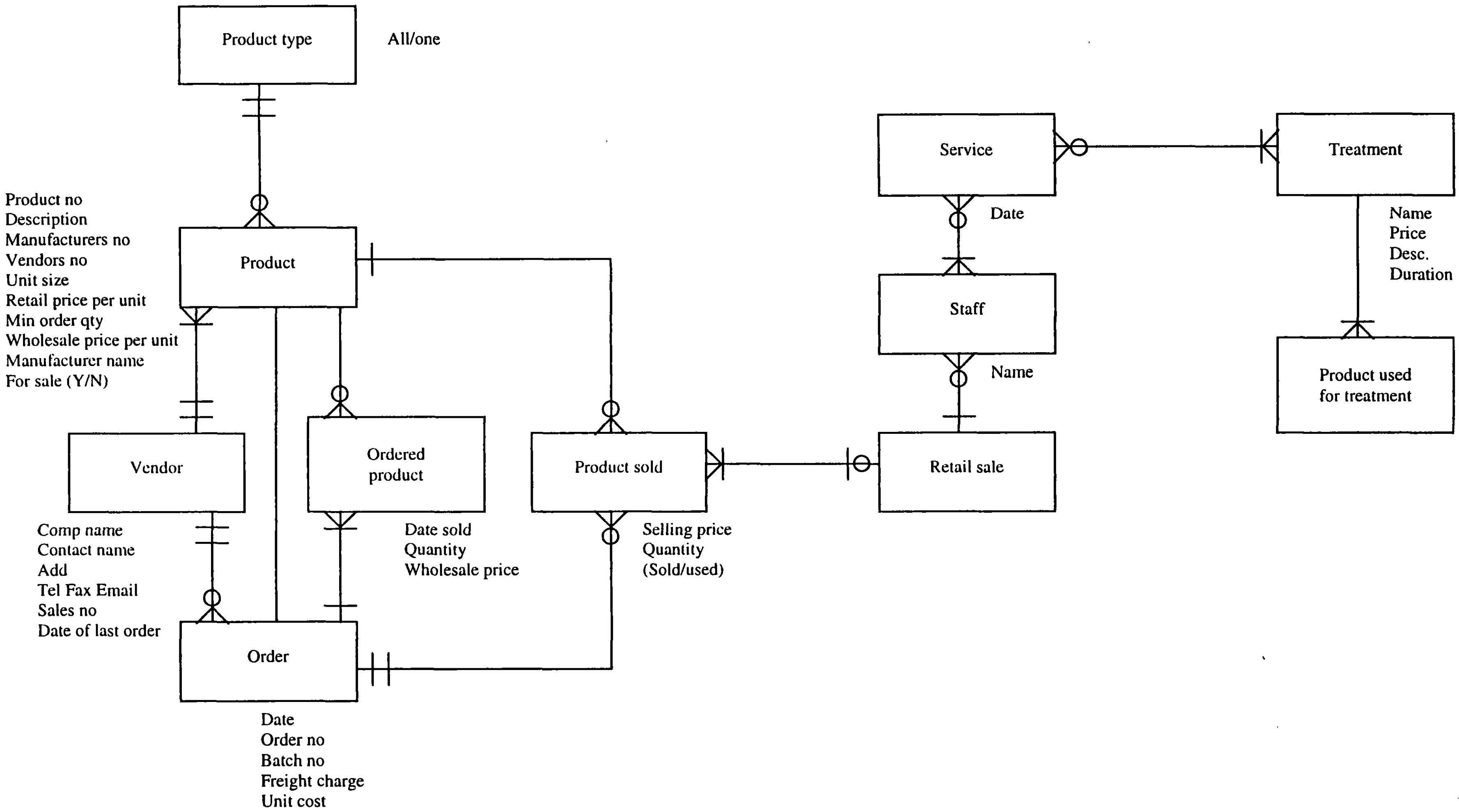
A Name is a text value
An email address is a text value
A Telephone No. is a number
An Address Line 1 is a text value
An Address Line 2 is a text value
An Address Line 3 is a text value
A City is a text value
A Postal Code is a text value
An Identifier is a number
A Method Description is a text value
A PR Impact is a text value
A Legal Impact is a text value
A Financial Impact is a number
A Process Description is a text value
A Risk is a text value
An Implementation Cost is a number
A Failure Cost is a number
A Failure Likelihood % is a number
An Annual Loss Expectancy is a number
A Protocol Description is a text value
A GSM Entity Description is a text value
An Issue Description is a text value
A Date & Time is a date/time
A Publish Date is a date/time
An Algorithm Type Description is a text value
A Name is a text value
A Title is a text value
A Document Reference is a number
A Desired Result is a text value
An Outcome Description is a text value
A Deadline is a date/time
A Review Date is a date/time
An Evaluation is a text value
An Audit Result is a text value
A Certificate Number is a number
An Expiry Date is a date/time
A Fee is a number
An International Dialling Code is a number
A Description is a text value
A Start Date Time is a date/time
An End Date Time is a date/time
A Price is a number
A Standard Guaranteed is a text value

E.17 Model 17 (Security/Fraud)



SECURITY/FRAUD MODEL V 1.0

E.18 Model 18 (Stock Control)


Product type
All/one
Product no
Description
Manufacturers no
Vendors no
Unit size
Retail price per unit
Min order qty
Wholesale price per unit
Manufacturer name
For sale (Y/N)
Product
Vendor
Ordered product
Product sold
Retail sale
Comp name
Contact name
Add
Tel Fax Email
Sales no
Date of last order
Date sold
Quantity
Wholesale price
Selling price
Quantity
(Sold/used)
Order
Date
Order no
Batch no
Freight charge
Unit cost
Service
Treatment
Date
Staff
Name
Name
Price
Desc.
Duration
Product used for treatment

E.19 Model 19 (Theatrical Productions)

A student is a person

Each student has a first name
Each student has a surname
Each student may have productions

A team member is a person

Each team member has a first name
Each team member has a surname
Each team member may have roles
Each team member has a production (*)

A role is an activity

Each role has a team member
Each role may have tasks
Each role has a overall description

A task is an activity

Each task has a role (*)
Each task has a overall description
Each task has a deadline

A venue is a place

Each venue has a venue name
Each venue may have positions
Each venue may have production locs (*)

A production is an activity

Each production has a student
Each production may have team members
Each production has a title
Each production may have sound systems
Each production may have prod. locations

A position is a category

Each position has a venue (*)
Each position has a overall description

A lighting circuit no is a system

Each lighting circuit no has a circuit no
Each lighting circuit no has a dimmer no
Each lighting circuit no may have lanterns
Each ltg. circuit no may have dim. states (*)

A lantern type is a category

Each lantern type has a overall description
Each lantern type may have lanterns (*)

A gobo is an object

Each gobo has a overall description
Each gobo has a lantern (*)

A gel is an object

Each gel has a lee/rosco no
Each gel has a lantern (*)

A direction is an activity

Each direction has a overall description
Each direction may have lanterns (*)

A lighting state is an object

Each lighting state has a description
Each lighting state has a cue
Each lighting state has a cue no
Each lighting state has a lighting plan (*)
Each lighting state may have dimmer states

A lighting plan is an activity

Each lighting plan has a overall description
Each lighting plan may have lighting states
Each lighting plan has a production location

A sound system is an activity

Each sound system has a production (*)
Each sound system has a description
Each sound system may have sfx cues

A sfx cue is an object

Each sfx cue has a cue
Each sfx cue has a cue no
Each sfx cue has a sound system (*)
Each sfx cue has a player
Each sfx cue has a sfx
Each sfx cue has a format type

A production location is a place

Each production location has a venue
Each production location has a production
Each production location has a lighting plan
Each production location may have perf. runs

A performance run is an activity

Each performance run has a perf. start date
Each performance run has a prod. location
Each performance run has a perf. end date

A lantern is an object

Each lantern has a lighting circuit no (*)
Each lantern has a lantern type

Each lantern may have gobos
Each lantern may have gels
Each lantern has a direction

A dimmer state is an activity

Each dimmer state has a lighting circuit no
Each dimmer state has a lighting state (*)
Each dimmer state has a dimmer level

A first name is a text value
A surname is a text value
An overall description is a text value
A deadline is a date/time
A venue name is a text value
A title is a text value
A performance start date is a date/time
A circuit no is a number
An alternatives is a text value
A lee/rosco no is a number
A description is a text value
A cue is a text value
A cue no is a number
A dimmer no is a number
A player is a number
A sfx is a text value
A format type is a text value
A performance end date is a date/time
A dimmer level is a number

Bibliography

Ackoff R (1967) 'Management misinformation systems', *Management Science*, 14 (4), B147-B156.

Aitkenhead A M, Slack J M (1985) *Issues in Cognitive Modelling: A Reader*, Erlbaum in association with the Open University.

Allport D A, Antonis B, Reynolds P (1972) 'On the division of attention: A disproof of the single channel hypothesis', *Quarterly Journal of Experimental Psychology*, 24, 225-35.

Andrews D C, Leventhal N S (1992) *Fusion: Integrating IE, CASE, and JAD: A Handbook for Reengineering the Systems Organization*, Yourdon Press.

Angelaccio M, Tiziana C, Santucci G (1990) 'QBD*: A graphical query language with recursion', *IEEE Transactions on Software Engineering*, 10 (16), October 1990.

Ashcraft M H (1998) *Fundamentals of Cognition* (1st Edition), Addison-Wesley.

Ashworth C, Goodland M (1990) *SSADM: A Practical Approach*, McGraw-Hill.

Ashworth C, Slater L (1992) *An Introduction to SSADM Version 4*, McGraw-Hill.

Atkinson M, Bacilhon F, DeWitt, D, Dittrich, K, Maier, D, Zdonik, S (1989) 'The object-oriented database system manifesto', *Proceedings of the First International Conference on Deductive and Object-Oriented Databases, Kyoto*, 40-57.

Atkinson P (1992) *Understanding Ethnographic Texts,* SAGE Publications.

Atwood T (1989) 'An introduction to object-oriented database management systems', *Hotline on Object-Oriented Technology*, 1 (1), SIGS Publications Inc, New York, 11-12.

August J (1990) *Joint Application Design: The Group Session Approach to System Design*, Prentice Hall.

August J H (1991) *Joint Application Design: The Group Session Approach to System Design,* Yourdon Press/Prentice-Hall International.

Auramaki E, Lehtinen E, Lyytinen K (1988) 'A speech-act-based office modelling approach', *ACM Transactions on Office Information Systems*, 6 (2).

Avery G, Baker E (1990) *Psychology at Work*, Prentice-Hall.

Avison D E (1997) 'The discipline of information systems: teaching, research, and practice', in J Mingers, F Stowell (eds.) *Information Systems: An Emerging Discipline?*, McGraw-Hill.

Avison D E, Fitzgerald G (1996) *Information Systems Development: Methodologies, Techniques, and Tools* (2nd ed.) Blackwell.

Avison D E, Myers M D (1995) *The Interdisciplinary Nature of Information Systems and the Case for Anthropology*, Discussion papers in Accounting and Management Science, Department of Accounting and Management Science, University of Southampton.

Avison D E, Nandhakumar J (1995) 'The discipline of information systems: let many flowers bloom!', in E D Falkenberg, W Hesse and A Olive (eds) *Information System Concepts Towards a Consolidation View*, Chapman & Hall.

Avison D E, Wood-Harper A (1990) *Multiview: an Exploration of Information Systems Development*, Blackwell Scientific.

Avison D E, Wood-Harper A T, Vidgen R T, Wood J R G (1996) 'Multiview 2: A further exploration into information systems development', *Proceedings of the 6th Annual BIT Conference, 'Business Information Systems - Uncertain Futures'*, Manchester Metropolitan University, 7th November, 295-303.

Backhouse J (1988) 'Simpler and richer cognitive models for ISAD', *Proceedings of the IFIP Working Group 8.1 International Workshop on Human Factors of Information Systems Analysis and Design (IFIP WHISAD '88)*, London, United Kingdom, July 28-29.

Baddeley A (1986) *Working Memory*, Claredon Press.

Bailin S C (1989) 'An object-oriented requirements specification method', *Communications of the ACM*, 32 (5), 608-623.

Bandura A (1986) *Social Foundations of Thought and Action: A Social Cognitive Approach*, Prentice-Hall.

Banerjee J et al (1987) 'Data modelling issues for object-oriented applications', *ACM Transactions on Office Information Systems*, 5 (1), 3-26.

Bansler J P, Bødker K (1993) 'A reappraisal of structured analysis: design in an organisational context', *ACM Transactions on Information Systems*, 11 (2), 165-193.

Barker R (1990) *Case Method: Entity Relationship Modelling*, Addison-Wesley.

Barker R (1992) 'The future for open standards in CASE', *Professional Computing (Australia)*, February 1992, 2-8.

Barlow H, Blakemore C, Weston-Smith M (1990) *Images and Understanding: Thoughts about Images, Ideas about Understanding*, Cambridge University Press.

Barlow J F (1990) 'Putting information systems planning methodologies into perspective', *Journal of Systems Management*, July 1990.

Bartiis S, Szymanski K, Harkins S G (1988) 'Evaluation and performance: a two-edged knife', *Personality and Social Psychology Bulletin*, 14, 242-51.

Baskerville R, Myers M, Nielsen P A, Wood-Harper T (1997) 'The impact of action research on information systems', *Information Systems and Qualitative Research*, A Lee, J Liebenau, J DeGross (eds.) p 69, Chapman & Hall.

Baskerville R, Wood-Harper A T (1996) 'A critical perspective on action research as a method for information systems research', *Journal of Information Technology*, 11 (3).

Baskerville R, Wood-Harper A T (1998) 'Diversity in information systems action research methods', *European Journal of Information Systems* 1998.

Battaglia G (1991) 'Strategic information planning: a corporate necessity', *Journal of Systems Management*, February, 23-26.

Beath C M, Orlikowski W (1994) 'The contradictory nature of systems development methodologies: deconstructing the information systems-user relationship in information engineering', *Information Systems Research*, 5 (4) 350-377.

Bechtel W, Abrahamsen A (1991) *Connectionism and the Mind: An Introduction to Parallel Processing in Networks*, Blackwell.

Beer S (1985) *Diagnosing the System for Organisations*, Wiley.

Bellezza F S (1983) 'The spatial arrangement mnemonic', *Journal of Education Psychology*, 75, 830-7.

Bemelmans M A (ed.) (1984) *Beyond Productivity: Information Systems Development for Organisational Effectiveness*, North-Holland.

Benyon D (1997) *Information and Data Modelling*, McGraw Hill.

Berger P L, Luckman T (1966) *The Social Construction of Reality: A Treatise in the Sociology of Knowledge*, Anchor Books.

Berrisford T R, Wetherbe J C (1979) 'Hueristic development: a redesign of systems design', *MIS Quarterly,* 3 (1), 11-20.

Best J B (1999) *Cognitive Psychology* (5th ed.), Wadsworth.

Bicarregui J C, Fitzgerald J S, Lindsay P A, Moore R, Ritchie B (1994) *Proof in VDM: A Practitioner's Guide*, Springer-Verlag.

Bijker W E, Law J (eds.) (1992) *Shaping Technology: Building Society Studies in Sociotechnical Change,* MIT Press.

Billings C, Billings M, Tower J (1997) *Rapid Application Development with Oracle Designer/2000*, Addison Wesley.

Blaha M R, Premerlani W J, Rumbaugh J E (1988) 'Relational database design using an object-oriented methodology', *Communications of the ACM*, 31, 4, April 1988.

Blum B I (1990) 'Human information processing in software development', *Human Factors in Information Systems Analysis and Design*, A Finkelstein, M Tauber, R Traunmuller (eds.) North-Holland.

Boar B H (1984) *Applications Prototyping: A Requirements Definition Strategy for the 80's*, Wiley.

Boland R J (1985) 'Phenomenology: a preferred approach to research on information systems', *Research Methods in Information Systems*, E Mumford et al (eds.) North-Holland.

Boland R J Jr (1991) 'Information system use as a hermeneutic process', Nissen H E, Klein H K, Hirschheim R (eds.) *Information Systems Research: Contemporary Approaches and Emergent Traditions, Proceedings of the IFIP TC8/WG8.2 Working Conference on the Information Systems Research Arena of the '90s Challenges, perceptions and Alternative Approaches, Copenhagen, Denmark, 14-16 December, 1990*, North-Holland.

Boland R J, Day W (1982) 'The process of system design: a phenomenological approach', *Proceedings of the 3rd International Conference on I.S,* Ann Arbor, Michigan, M Ginzberg, C Ross (eds.) 31-45.

Bollinger T B, Pfleeger S L (1990) 'The economics of reuse: issues and alternatives', *Proceedings of the 8th Annual National Conference on Ada Technology.*

Booch G (1991) *Object-Oriented Design with Applications*, Benjamin Cummings.

Booch G (1996) *Object Solutions: Managing the Object-Oriented Project*, Addison-Wesley.

Booch G, (1986) 'Object-oriented development', *IEEE Transactions on Software Engineering*, Vol SE-12, February 1986.

Bostrom, Heinen (1977) 'MIS problems and failures: a socio-technical perspective', *MIS Quarterly*, 1 (3), 17-32, (4), 11-25.

Bourne L, Dominowski R L, Loftus E F (1979) *Cognitive Processes*, Prentice-Hall.

Bower G H, Clark M C, Lesgold A M, Winzenz D (1969) 'Hierarchical retrieval schemes in recall of categorised word lists', *Journal of Verbal Learning and Verbal Behaviour*, 8, 232-43.

Brandimonte M, Hitch G H, Bishop D V M (1992) 'Influence of short-term memory codes on visual image processing: evidence of image transformation tasks', *Journal of Experimental Psychology: Learning, Memory and Cognition*, 18, 157-165.

Briggs T L, Werth J (1994) 'A specification language for object-oriented analysis and design', M Tokoro, R Pareschi (eds.) *Proceedings ECOOP'94, Bologna, Italy, July 1994*, Springer-Verlag, 365-385.

Brinkkemper S, ter Hofstede A H M (1989) *The Conceptual Task Model: a Specification Technique Between Requirements Engineering and Program Development*, Technical Report no 89-15, Department of Informatics, University of Nijmegen, Netherlands.

Brodie M L, Silva E (1982) 'Active and passive component modelling: ACM/PCM', Olle T W et al (eds.) *Information System Design Methodologies: A Comparative View*, North Holland.

Brosnan M J (ed.) (1996) *Cognitive Functions: Readings in Representation and Reasoning*, School of Social Sciences, University of Greenwich.

Brown R (1988) *Group Processes: Dynamics Within and Between Groups*, Blackwell.

Bryant A (1993) 'Development of the SSADM standard', *Proceedings of SESS 1993*, Brighton, UK.

Bryant A (1994) *British Standard for the Development of Information Systems Products Using SSADM (BS7738),* British Standards Institute.

Bryant A (1995) *Standardising SSADM: Methods, Standards and Maturity,* McGraw Hill.

Bryman A (1989) *Research Methods and Organisation Studies*, Unwin Hyman.

Bubenko J A (1986) 'Information system methodologies—a research view', *Information Systems Design Methodologies: Improving the Practice*, Olle T W et al (eds.) North-Holland.

Bubenko J, Rolland C, Loucopoulos P, deAutonellis V (1994) 'Facilitating "fuzzy to formal" requirements engineering', *Proceedings of the IEEE International Conference on Requirements Engineering*, 154-7.

Bullinger H J et al (eds.) (1988) *Information Technology for Organisational Systems: Concepts for Increased Competitiveness* (Proceedings of the First European Conference on Information Technology for Organisational Systems—EURINFO '88), Athens, Greece, May, North-Holland.

Bunzl M (1996) 'The logic of thought experiments', *Syntheses*, 106 (2), 227-240.

Burgess A (1992) *A Mouthful of Air: Language, Languages - Especially English*, Vintage.

Burnstein E, Vinokur A (1975) 'What a person thinks upon learning he has chosen differently from others: nice evidence for the persuasive-arguments explanation of choice shifts', *Journal of Experimental Social Psychology*, 11, 412-426.

Burrell G, Morgan G (1979) *Sociological Paradigms and Organizational Analysis*, Heinemann.

Campbell D T, Fiske D W (1959) 'Convergent and discriminant validation by the multitrait-multimethod matrix', *Psychological Bulletin*, 56 (2) 81-105.

Campbell J P (1985) 'Editorial: some remarks from the outgoing editor', Cummings LL, Pennings J M (eds.) *Publishing in the Organisational Sciences*, Jossey-Bass.

Cardenas A, McLeod D (eds.) (1990) *Research Foundations in Object-Oriented and Semantic Database Systems*, Prentice Hall.

Carlson (1979) 'Business information analysis and integration technique (BIAIT): the new horizon', *Data Base*, 10 (4), 3-9.

Cassell C, Symon G (1994) 'Qualitative research in work contexts', Cassell C, Symon G (eds.) *Qualitative Methods in Organisation Research*, Sage Publications.

Cattell R G G (1991) *Object Data Management: Object-Oriented and Extended Relational Database Systems*, Addison Wesley.

Chang M T, Woo C C (1994) 'A speech-act based negotiation protocol: design, implementation and test use', *ACM Transactions on Information Systems*, 12 4, 360-382.

Chang S K (1990) A visual language compiler for information retrieval by visual reasoning', *IEEE Transactions on Software Engineering*, 10 (16), October.

Checkland P B, Scholes J (1990) *Soft Systems Methodology in Action*, Wiley.

Checkland, P (1981) *Systems Thinking, Systems Practice*, John Wiley, Chichester.

Chen P (1976) 'The Entity Relationship Model - toward a unified view of data', *ACM Transactions on Database Systems*, 1 (1), 9-36.

Chen T L C, Sutton M M (1988) 'Object-oriented design: is it enough for large Ada systems?', *Proceedings of the 8th Annual National Conference on Ada Technology*.

Chomsky N A (1968) 'Language in the mind', Cashdan et al (eds.) *Language in Education: A Source Book*, Language and Learning Course Team, Open University, Routledge & Keegan Paul.

Ciborra C U (ed.) (1996) *Groupware and Teamwork: Invisible Aid or Technical Hindrance*, John Wiley.

Ciborra C U, Jelassi T (1994) *Strategic Information Systems: A European Perspective,* Wiley.

Clark, P A (1972) *Action Research and Organisational Change*, Harper and Row.

Clegg D, Barker R (1994) *CASE Method Fast-Track: A RAD Approach,* Addison-Wesley.

Coad P (1990) 'Tutorial: object-oriented analysis', *Technology of Object-Oriented Languages and Systems: TOOLS Pacific '90*, Sydney, Australia.

Coad P (1991) 'New advances in object-oriented analysis', *Journal of Object-Oriented Programming*, January.

Coad P, Yourdon E (1991) *Object-Oriented Analysis* (2nd ed.) Yourdon Press.

Coad P, Yourdon E (1991) *Object-Oriented Design*, Yourdon Press.

Codd E F (1990) *The Relational Model for Database Management,* Addison-Wesley.

Colbert E (1989) 'The object-oriented software development method: a practical approach to object-oriented development', *Proceedings of TRI-Ada '89*, October.

Coleman D, Hayes F (1991) 'Getting the best from objects: the experience of HP', *Proceedings of the IEE Colloquium on Applications and Experience of Object-Oriented Design*, January.

Collins A M, Quillian M R (1969) 'Retrieval time from semantic memory', *Journal of Verbal Learning and Verbal Behaviour*, 8, 240-8.

Collins A, Loftus E (1975) 'A spreading activation theory of semantic processing', *Psychological Review*, 82, 407-428.

Colter M (1984) 'A comparative analysis of systems analysis techniques', *MIS Quarterly*, 8 (1), 51-66.

Connell J L, Shafer L (1995) *Object-Oriented Rapid Prototyping*, Yourdon Press.

Cooper R B, Swanson E B (1979) 'Management information requirements assessment: the state of the art', *Data Base*, 11 (2), Fall 1979, 5-16.

Crabtree B F, Miller W L (1992) 'A template approach to text analysis: developing and using codebooks', Crabtree B F, Miller W L (eds.) *Doing Qualitative Research,* Sage.

Craik F I M, Lockhart R S (1972) 'Levels of processing: a framework for memory research', *Journal of Verbal Learning and Verbal Behaviour,* 11, 671-84.

Craik F I M, Tulving E (1975) 'Depth of processing and the retention of words in episodic memory', *Journal of Experimental Psychology: General*, 104, 268-94.

Crawford A (1993) *Advancing Business Concepts in a JAD Workshop Setting: Business Engineering and Process Redesign*, Prentice Hall.

Curtice B, Stringer D (1991) 'Visualizing information planning: three strategic techniques that rely on graphics, not jargon', *Computerworld*, January 14th 1991, 63-66.

Cutts G (1988) *SSADM: Structured Systems Analysis and Design Methodology*, Van Nostrand.

D'Haenens I et al (1990) 'Experiences with rule-based dynamic modelling', *Dynamic Modelling of Information Systems*, Sol H G, van Hee K M (eds.) North-Holland.

Davenport T (1993) *Process Innovation: Reengineering Work through Information Technology,* Harvard Business School Press.

Davenport T H (1990) *The New Industrial Engineering: Information Technology and Business Process Redesign*, Working Paper No. 213, Centre for Information Systems Research, Sloan School of Management, Massachusetts Institute of Technology.

Davies L J (1991) 'Assessing the business impact of information systems failures: a risk analysis methodology', *Computer Control Quarterly*, 9 (4), Oct 1991.

Davies L, Wood-Harper A T (1990) 'Multiple perspectives in desk-top computing', *Desktop Information Technology*, Kaiser K M, Oppelland H J (eds.) North-Holland.

Davis A M (1988a) 'A taxonomy of the early stages of the software development life cycle', *Journal of Systems and Software*, 8, 297-311.

Davis A M (1988b) *Software Requirements: Analysis and Specification*, Prentice-Hall.

Davis, G B (1987) 'Strategies for information requirements determination', *Information Analysis: Selected Readings*, R Galliers (ed.) Addison Wesley.

Dawes J (1991) *The VDM Reference Guide*, Pitman.

Day J C, Bellezza F S (1983) 'The relation between visual imagery mediators and recall', *Memory and Cognition*, 11, 251-7.

de Champeaux D (1991) *A Comparative Study of Object-Oriented Analysis Models (Technical Report HPL-01-41)*, Hewlett Packard Laboratories, Palo Alto CA.

de Champeaux D (1991b) 'Object-oriented analysis and top-down software development', *Proceedings of 5th European Conference on Object-Oriented Programming (ECOOP '91)*, Springer Verlag, 360-376.

Dean H (1991) 'Object-oriented design using message flow decomposition', *Journal of Object-Oriented Programming*, May 1991, 21-31.

Delbeq A, Van de Ven A, Gustafson D (1975) *Group Techniques for Program Planning*, Scott Foresman.

DeMarco T, Lister T (1987) *Peopleware: Productive Projects and Teams*, Dorset House.

Denzin N (1970) *The Research Act in Sociology*, Butterworth.

Diederich J, Milton J (1989) 'Objects, messages and rules in database design', *Object-Oriented Concepts, Databases and Applications*, Kim W, Lochovsky H (eds.) ACM Press, New York 1989.

Dilts R B, Grinder J, Bandler R, DeLozier J, Cameron-Bandler L (1979) *Neuro-Linguistic Programming* (Volume I), Meta Publications.

Doyle, K G, Wood, J R G, Wood-Harper A T (1993) 'Soft systems and systems engineering: on the use of conceptual models in information systems development', *Journal of Information Systems*, 3, 187-198.

Duke R, King P, Rose G, Smith G (1991) *The Object-Z Specification Language, Version 1*, Technical Report 91-1, Software Verification Research Centre, University of Queensland.

Dyer W G (Jr), Wilkins A L (1991) 'Better stories, not better constructs, to generate better theory: a rejoinder to Eisenhardt, *Academy of Management Review*, 16 (3), 613-619.

Eder J et al (1987) 'A behaviour design methodology for information systems', *Proceedings of the Phoenix Conference on Computers and Communications*, Arizona, 328-334.

Einsten A (1949) 'Autobiographical notes', Schlipp A (ed.) *Albert Einstein: Philosopher-Scientist*, Open Court.

Elan J J, Walz D B, Curtis B, Krasner H (1991) 'Measuring group process in software design teams', Nissen H E, Klein H K, Hirschheim R (eds.) *Information Systems Research: Contemporary Approaches and Emergent Traditions, Proceedings of the IFIP TC8/WG8.2 Working Conference on the Information Systems Research Arena of the '90s Challenges, Perceptions and Alternative Approaches, 14-16 December, 1990*, Copenhagen, Denmark, North-Holland.

Embley D W, Woodfield S N (1987) 'Cohesion and coupling for abstract data types', *Proceedings of the Phoenix Conference on Computers and Communications*, Arizona 229-234.

Essink L J B (1988) 'A conceptual framework for information systems development methodologies', *Information Technology for Organisational Systems*, H J Bullinger et al (eds.) North-Holland.

Eysenck M W (1990) *Cognitive Psychology: A Student's Handbook*, Lawrence Erlbaum Associates.

Eysenck M W, Keane M T (1995) *Cognitive Psychology: A Student's Handbook* (3rd edition), Lawrence Erlbaum.

Falkenberg E D et al (1983) 'Feature analysis of ACM/PCM, CIAM, ISAC and NIAM', T W Olle, H G Sol C J Tully (eds.) *Information Systems Design Methodologies: Feature Analysis, Proceedings of the CRIS '83 Conference*, North-Holland.

Falkenberg E, Lindgreen P (eds.) (1989) *information systems Concepts: An Indepth Analysis*, Proceedings of the IFIP Working Conference, Namur, North Holland.

Finkelstein A, Tauber M, Traunmuller R (1990) 'Understanding human action in information systems development', *Human Factors in Information Systems Analysis and Design*, Finkelstein A, Tauber M, Traunmuller R (eds.) North-Holland.

Finkelstein C (1990) *An Introduction to Information Engineering: From Strategic Planning to Information Systems*, Addison Wesley.

Fitzgerald G (1991) 'Validating new information systems techniques: a retrospective analysis', Nissen H E, Klein H K, Hirschheim R (eds.) *Information Systems Research: Contemporary Approaches and Emergent Traditions, Proceedings of the IFIP TC8/WG8.2 Working Conference on the Information Systems Research Arena of the '90s Challenges, perceptions and Alternative Approaches, 14-16 December, 1990*, Copenhagen, Denmark, North-Holland.

Floyd C (1986) 'A comparative evaluation of system development methods', *Information Systems Design Methodologies: Improving the Practice*, Olle T W et al (eds.) North-Holland 1986.

Floyd C, Reisin, F, Schmidt G (1989) 'STEPS to software development with users', Ghezzi C, McDermid J A (eds.) *ESEC'89 - 2nd European Software Engineering Conference*, University of Warwick, Coventry, Lecture Notes in Computer Science No 387, Springer-Verlag, 48–64.

Foote Whyte W (1991) *Participatory Action Research*, Sage.

Forster N (1994) 'Analysing company documentation', Cassell C, Symon G (eds.) *Qualitative Methods in Organisation Research*, Sage Publications.

Fowler M, Scott K (1997) *UML Distilled: Applying the Standard Object Modelling Language*, Addison-Wesley.

Frank H, Gerteis W (1990) 'Including dynamic modelling into object-oriented design', *Dynamic Modelling of Information Systems*, Sol H G, van Hee K M (eds.) North-Holland.

Frohlich D, Pekruhl U (1996) *Direct Participation and Organisational Change: Fashionable but Misunderstood? An Analysis of Recent Research in Europe, Japan, and the USA*, Office for Official Publications of the European Communities.

Gabbay D, McBrien P (1991) 'Temporal logic and historical databases', *Proceedings of the 17th International Conference on Very Large Databases*, Barcelona, September.

Gable G, 'Integrating Case Study and Survey Research Methods: An Example in Information Systems', *European Journal of Information Systems,* 3 (2), 112-126.

Galliers R (ed.) (1992) *Information Systems Research: Issues, Methods, and Practical Guidelines*, Blackwell Scientific Publications.

Galliers R D (1994) 'Relevance and rigour in information systems research: some personal reflections on issues facing the information systems research community', in Glasson B C et al (eds.) *Business Process Re-Engineering, Proceedings of the IFIP TC8 Open Conference on Business Process Re-Engineering*, Queensland, Australia, 1994, pp. 93-101.

Gane C, Sarson T (1979) *Structured Systems Anaysis: Tools and Techniques*, Prentice-Hall.

Garcia L, Quek F (1997) 'Qualitative research in information systems: time to be subjective?', A S Lee, J Liebenau, J I DeGross (Eds.) *Information Systems and Qualitative Research, Proceedings of the IFIP TC8 WG 8.2 International Conference on Information Systems and Qualitative Research 31st May-3rd June 199, Philadelphia, Pennsylvania, USA,* Chapman & Hall.

Garfunkel J (1990) 'COBOL—the next stage', *Computerworld*, July 23, 87-91.

Gibson E (1990) 'Objects—born and bred', *Byte*, October, 245-254.

Gibson J J (1986) *The Ecological Approach to Visual Perception*, Lawrence Erlbaum.

Gilhooly K J (1982) *Thinking: Directed, Undirected and Creative*, Academic Press.

Goodland M, Slater (1995) *SSADM - Version 4: A Practical Approach* (2nd Edition), McGraw Hill.

Gould S J (1996) *The Mismeasure of Man,* WW Norton and Co.

Grabner R (1998) *The Remote Model Displaying Tool: A Java Application*, Final Year BA Project Thesis, Department of Computer Science, Trinity College Dublin.

Green D W (ed.) (1996) *Cognitive Science: An Introduction*, Blackwell.

Gregory F H (1993) 'Logic and meaning in conceptual models: implications for information system design', *Systemist,* 15 (1), 28-43.

Gregory F H (1993) 'Mapping conceptual models onto the real world', *Systems Science: Addressing Global Issues*, F A Stowell, D West and J G Howell (eds.) 117-122, Plenum Press.

Gregory F H (1995) *Mapping Information Systems Onto the Real World*, City University of Hong Kong, Department of Information Systems, Working Paper No. WP 95/01.

Gregory R L (1966) *Eye and Brain: The Psychology of Seeing*, McGraw-Hill.

Gregory R L (1972) 'Cognitive contours', *Nature*, 238, 51-2.

Guinan P J, J G Cooprider, S Sawyer (1997) 'The effective use of automated application development tools', *IBM Systems Journal*, 36(1).

Guy A (1998) Private communication.

Habermas J (1972) *Knowledge and Human Interests*, Heinemann.

Hackathorn R D, Karimi J (1988) 'A framework for comparing information engineering methods', *MIS Quarterly*, June 1988, 203-220.

Hall J, Williams M A (1966) 'A comparison of decision-making performance in established and ad hoc groups', *Journal of Personality and Social Psychology*, 3, 214-222.

Halpin T A, Nijssen G M (1989) *Conceptual Schema and Relational Database Design: a Fact-Oriented Approach*, Prentice Hall.

Halpin T A, Nijssen G M (1995) *Conceptual Schema and Relational Database Design* (2nd ed.) Prentice Hall.

Hammer M, Champy J (1993) *Reengineering the Corporation: a Manifesto for Business Revolution*, Nicholas Brealey.

Hammersley M (1992) *What's Wrong with Ethnography: Methodological Explorations,* Routledge.

Hanseth O, Moneiro E (1994) 'Modelling and the representation of reality: some implications of philosophy on practical systems development', *Scandinavian Journal of Information Systems*, 6 (1) 47-68.

Harel D (1987) 'Statecharts: a visual formalism for complex systems', *Scientific Computer Programming,* 8 (3), 231-274.

Harvey L (1997) 'A discourse on ethnogrphy', Lee A S, Liebenau J, DeGross J I (eds.) *Information Systems and Qualitative Research: Proceedings of the IFIP TC8 WG8.2 International Conference on Information Systems and Qualitative Research, 31st May-3rd June 1997,* Philadelphia, Pennsylvania, USA, Chapman and Hall.

Hawkins J M (1988) *The Oxford Paperback Dictionary* (3rd ed.) Oxford University Press.

Hay D C (1996) *Data Model Patterns: Conventions of Thought*, Dorset House.

Headland T N, Pike K L, Harris M (1990) *Emics and Etics: The Insider/Outside Debate*, Sage Publications.

Hellenack L J (1997) 'Object-Oriented Business Patterns, *Object Magazine*, January 1997.

Henderson-Sellers B (1992a) 'Object-oriented information systems: an introductory tutorial', *Australian Computer Journal*, 1 (24), February, 12-24.

Henderson-Sellers B (1992b) *A Book of Object-Oriented Knowledge*, Prentice Hall.

Henderson-Sellers B, Constantine L L (1991) 'Object-oriented development and functional decomposition', *Journal of Object-Oriented Programming*, January.

Henderson-Sellers B, Edwards J (1994) *Book Two of Object-Oriented Knowledge: The Working Object*, Prentice Hall.

Henderson-Sellers B, Edwards J M (1990) 'The object-oriented systems life cycle', *Communications of the ACM*, 33 (9).

Herbst H (1997) *Business Rule-Oriented Conceptual Modeling*, Springer-Verlag.

Herndon K P, van Dam A, Gleicher M (1994) 'Workshop report: the challenges of 3D interaction', *ACM SIGCHI Bulletin*, October.

Hirakawa M, Iwata S, Tahara Y, Tanaka Y, Ichikawa T (1988) 'A framework for construction of icon systems', *Proceedings of the 1988 IFIP Workshop on Visual Languages*, 70-77.

Hirschheim R A (1985) *Office Automation: A Social and Organisational Perspective*, Wiley.

Hirschheim R A, Klein H K, Lyytinen K (1995) *Information systems development and data modelling: conceptual and philosophical foundations*, Cambridge University Press.

Hitchman S (1997) 'Using DEKAF to understand data modelling in the practitioner domain', *European Journal of Information Systems*, 6 (3).

Hofman J D (1995) *Radical Change in the Systems Delivery Process: Object-Oriented Component Assembly*, Centre for information systems Research, Sloan School of Management, MIT.

Hofman J D, Rockart J F (1994) *Application Templates: Faster, Better, Cheaper Systems*, Sloan Management Review, 36 (1), Fall 1994, 49-60.

Holsti E R (1971) 'Crisis, stress, and decision-making', *International Social Science Journal*, 23, 53-67.

Holyoak K J, Thagard P (1995) *Mental Leaps: Analogy in Creative Thought*, MIT Press.

Hong S, Maryanski F (1990) 'Representation of object-oriented data models', *Information Sciences* 52, 247-284.

Howard P, Butler J, Barrie, S (1997) *Fast Development Software: An Evaluation and Comparison of RAD Tools*, Bloor Research.

Huizing C, de Roever W P (1991) 'Introduction to design choices in the semantics of statecharts', *Information Processing Letters*, 37, 205-213.

IBM (1981) *Business Systems Planning Manual #GE20-0527-3*, IBM Corporation.

Jacobson I (1987) 'Object-oriented development in an industrial environment', *Proceedings of the Conference on Object-Oriented Systems, Languages and Applications (OOPSLA '86)*, ACM.

Jacobson I (1991) 'Industrial development of software with an object-oriented technique', *Journal of Object-Oriented Programming*, March-April 1991.

Jalote P (1989) 'Functional refinement and nested objects for object-oriented design', *IEEE Transactions on Software Engineering*, 15 (3), 264-270.

Janis I (1972) *Victims of Groupthink*, Houghton Mifflin.

Janis I, Mann L (1977) *Decision Making*, Free Press.

Janson M, Guimaraes T, Brown A, Taillieu T (1997) 'Exploring a chairman of the baord's construction of organizational reality: the Colruyt case', Lee A S, Liebenau J, DeGross J I (eds.) *Information Systems and Qualitative Research: Proceedings of the IFIP TC8 WG8.2 International Conference on Information Systems and Qualitative Research, 31st May-3rd June 1997,* Philadelphia, Pennsylvania, USA, p.69, Chapman and Hall.

Jayaratna N (1994) *Understanding and Evaluating Methodologies*, McGraw-Hill.

Johnson G I, Briggs P (1994) 'Question-asking and verbal protocol techniques', Cassell C, Symon G (eds.) *Qualitative Methods in Organisation Research*, Sage Publications.

Johnson R E, Foote B (1988) 'Designing reuseable classes', *Journal of Object-Oriented Programming*, June/July.

Johnson-Laird P N (1983) 'Deductive reasoning ability', *Journal of Experimental Psychology,* 112, 173-194.

Johnson-Laird P N (1983) *Mental Models: Towards a Cognitive Science of Language, Inference and Consciousness*, Cambridge University Press.

Johnson-Laird P N (1993) *The Computer and the Mind: An Introduction to Cognitive Science* (2nd ed.) Fontana.

Johnson-Laird P N (1994) 'Mental models and probabilistic thinking', *Cognition*, 50, 189-209.

Kappel G, Schrefl M (1988) 'A behaviour-integrated entity-relationship approach for the design of object-oriented databases', *Proceedings of the 7th International Conference on the Entity-Relationship Approach*, Roma, November,175-192.

Kappel G, Schrefl M (1990) 'Using an object-oriented diagram technique for the design of information systems', Sol H G, van Hee K M (eds.) *Dynamic Modelling of Information Systems*, North-Holland.

Keane M T, Gilhooly K J (eds.) (1992) *Advances in the Psychology of Thinking* (Vol 1), Harvester Wheatsheaf.

Keen P G W (1991) 'Relevance and rigor in information systems research: improving quality, confidence, cohesion and impact', in H-E Nissen, H K Klein and R Hirschheim (eds)

Information Systems Research: Contemporary Approaches and Emergent Traditions, North-Holland.

Keen P G, Scott Morton M S (1978) *Decision Support Systems: An Organisational Perspective,* Addison-Wesley.

Kellogg R T (1997) *Cognitive Psychology*, Sage Publications.

Kerner D V (1979) 'Business Information Characterisation Study', *Data Base*, Spring, 10-17.

Keyes J (1991) 'Unstructured objects focus of new analysis', *Software Magazine*, March, 40-50.

Khadem R, Blanchard K M (1989) *One Page Management: How To Use Information To Achieve Your Goals*, Quill Books.

Khadem R, Lorber R (1998) *One Page Management: Success Through Access! How To Get The Key Information You Need To Lead Your People,* Quill Books.

Khanna T (1990) *Foundations of Neural Networks*, Addison-Wesley.

Kiesler S, Siegel J, McGuire T (1984) 'Social psychological aspects of computer-mediated communication', *American Psychologist*, 39, 1123-34.

Kim W, Lochovsky H (1989) *Object-Oriented Concepts, Databases and Applications*, ACM Press.

King (1978) 'Strategic planning for management information systems', *MIS Quarterly*, 2 (1), 27-37.

King J L, Applegate L M (1997) 'Crisis in the case research crisis: marginal diminishing returns to scale in the qualitative-quantitative research debate', Lee A S, Liebenau J, DeGross J I (eds.) *Information Systems and Qualitative Research: Proceedings of the IFIP TC8 WG8.2 International Conference on Information Systems and Qualitative Research, 31st May-3rd June 1997,* Philadelphia, Pennsylvania, USA, p.69, Chapman and Hall.

King N (1994) 'The qualitative research interview', Cassell C, Symon G (eds.) *Qualitative Methods in Organisation Research*, Sage Publications.

King R (1989) 'My cat is object-oriented', Kim W, Lochovsky H (eds.) *Object-Oriented Concepts, Databases and Applications*, ACM Press.

King S F (1996) 'CASE tools and organisational action', *Journal of Information Systems*, 6, 173-194.

Kirk J, Miller M (1986) *Reliability and Validity in Qualitative Research*, Sage.

Kitaoka B J (1990) 'Repository support for a reuse process', *Proceedings of the 8th Annual National Conference on Ada Technology.*

Klas W, Neuhold E J, Schrefl M (1988) 'On an object-oriented data model for a knowledge base', Speth R (ed.) *Research into Networks and Distributed Applications*, North Holland, 1163-1175.

Köhler W (1947) *Gestalt Psychology* (Revised edition), Liveright.

Kolodner J L, Reisbeck (1986) *Experience, Memory, and Reasoning*, Lawrence Erlbaum.

Kosslyn S M (1994) *Image and Brain*, MIT Press.

Kudrass T, Lehmbach M, Buchmann A P (1996) 'Tool-based re-engineering of a legacy MIS: an experience report', Constantopoulos P, Mylopoulos J, Vassiliou Y (eds.) *Proceedings of the 8th International Conference on Advanced Information Systems Engineering CAiSE'96*, May 20-24, Heraklion, Crete, Springer-Verlag, 116-135.

Kuhn, J S (1970) *The Structure of Scientific Revolutions* (2nd Ed), University of Chicago Press.

Kurtz B et al (1991) *Object-Oriented Systems Analysis and Specification: A Model-Driven Approach*, Prentice-Hall.

Lachman R, Lachman J L, Butterfield, E C (1979) *Cognitive Psychology and Information Processing: An Introduction*, Lawrence Erlbaum.

Lakoff G (1987) *Women, Fire, and Dangerous Things: What Categories Reveal About the Mind*, University of Chicago Press.

Lamm H, Trommsdorf G (1973) 'Group versus individual performance on tasks requiring ideational proficiency (brainstorming): a review', *European Journal of Social Psychology*, 3, 361-388.

Latané B, Williams K, Harkins S (1979) 'Many hands make light the work: the causes and consequences of social loafing', *Journal of Personality and Social Psychology*, 37, 822-32.

Lau F (1997) 'A review on the use of action research in information systems studies', Lee A S, Liebenau J, DeGross J I (eds.) *Information Systems and Qualitative Research: Proceedings of the IFIP TC8 WG8.2 International Conference on Information Systems and Qualitative Research, 31st May-3rd June 1997,* Philadelphia, Pennsylvania, USA, p.69, Chapman and Hall.

Laurel B (1993) *Computers as Theatre* (2nd ed.), Addison-Wesley.

Lazarsfield P F (1958) 'Evidence and inference in social research', *Daedalus*, 87 (4) 99-130.

LBMS (1993) *Systems Engineer User Guide,* Learmonth and Burchett Management Systems Plc.

Lee A S, Liebenau J (1997) 'Information systems and qualitative research', Lee A S, Liebenau J, DeGross J I (eds.) *Information Systems and Qualitative Research: Proceedings of the IFIP TC8 WG8.2 International Conference on Information Systems and Qualitative Research, 31st May-3rd June 1997,* Philadelphia, Pennsylvania, USA, Chapman and Hall.

Lee A S, Liebenau J, DeGross J I (eds.) (1997) *Information Systems and Qualitative Research: Proceedings of the IFIP TC8 WG8.2 International Conference on Information Systems and Qualitative Research, 31st May-3rd June 1997,* Philadelphia, Pennsylvania, USA, Chapman and Hall.

Lee A S (1999) 'Researching MIS', in W Currie, B Galliers (Eds.) *Rethinking Management Information Systems: An Interdisciplinary Perspective*, Oxford University Press.

Lewis, P J (1993) 'Linking Soft Systems Methodology with data-focused information systems development', *Journal of Information Systems*, 3, 169-186.

Lindgreen P (ed.) (1990) *A Framework of Information Systems Concepts*, IFIP WG 8.1 Task Group Interim Report.

Loaisen J (1991) 'Pluralism or positivistic trivialism: important trends in contemporary philosophy of science', Nissen H E, Klein H K, Hirschheim R (eds.) *Information Systems Research: Contemporary Approaches and Emergent Traditions, Proceedings of the IFIP TC8/WG8.2 Working Conference on the Information Systems Research Arena of the '90s Challenges, perceptions and Alternative Approaches, 14-16 December, 1990*, Copenhagen, Denmark, North-Holland.

Loucopoulos P, Zicari R (1992) *Conceptual Modelling, Databases and CASE*, Wiley.

Lundeberg M (1982) 'The ISAC approach to specification of information systems and its application to the organisation of an IFIP working conference', T W Olle, H G Sol, A A Verrijn-Stuart (eds.) *Information Systems Design Methodologies: A Comparative Review.*

Lundeberg M, Goldkuhl G, Nilsson A (1981) *Information Systems Development: A Systematic Approach*, Prentice-Hall.

Lyytinen K (1987) 'Two views of information modelling', *Information and management*, 12 (1), 9-19.

Lyytinen K, Lehtinen E (1984) 'On information modelling through illocutionary logic', *Report of 3rd Scandinavian Research Seminar on Information Modelling and Database Management*, Kangassolo H (ed.) University of Tampere, Finland, 35-118.

Maddison R N (1983) *Information Systems Methodologies*, BCS/Wiley-Hayden.

Maddison R N et al (1984) 'A feature analysis of five information system methodologies', *Beyond Productivity: Information Systems for Organisational Effectiveness*, Bemelmans T H (ed.) North-Holland.

Maiden N (1991) 'Analogy as a paradigm for specification reuse', *Software Engineering Journal*, January, 3-15.

Malim T (1994) *Cognitive Processes: Attention, Perception, Memory, Thinking and Language*, Macmillan.

Manktelow K, Jones J (1987) 'Principles from the psychology of thinking and mental models', Gardiner M M, Christie B (eds.) *Applying Cognitive Psychology to User-Interface Design*, Wiley.

Marakas G M, Elam J J (1998) 'Semantic structuring in analyst acquisition and representation of facts in requirements analysis', *Information Systems Research,* 9 (1), 37-63.

Marcus M P (1980) *A Theory of Syntactic Recognition for Natural Language,* MIT Press.

Markus M L (1984) *Systems in Organisations: Bugs and Features*, Cambridge.

Markus M L (1997) 'The qualitative difference in information systems research and practice', Lee A S, Liebenau J, DeGross J I (eds.) *Information Systems and Qualitative Research: Proceedings of the IFIP TC8 WG8.2 International Conference on Information Systems and Qualitative Research, 31st May-3rd June 1997,* Philadelphia, Pennsylvania, USA, Chapman and Hall.

Markus M L, Bjorn-Andersen N (1987) 'Power over users: its exercise by system professionals', *Communications of the ACM*, 30 (6).

Marr D (1982) *Vision*, Freeman.

Martin J (1982) *Strategic Data Planning Methodologies*, Savant Institute.

Martin J (1990) *Information Engineering*, Prentice Hall.

Martin J (1993) *Extending Information Engineering with Rules and Objects* (white paper) James Martin & Co.

Martin J (1993) *Principles of Object Oriented Analysis and Design*, Prentice Hall.

Martin J, Odell J (1992) *Object-Oriented Analysis and Design*, Prentice-Hall.

Mathiassen L, Seewaldt T, Stage J (1995) 'Prototyping and specifying: principles and practices of a mixed approach', *Scandinavian Journal of Information Systems*, 7 (1).

Mathiassen L, Seewaldt T, Stage J (1995) 'Prototyping and specifying: principles and practices of a mixed approach', Scandinavian Journal of Information Systems, 7 (1).

Maude T, Willis G (1991) *Rapid Prototyping: The Management of Software Risk*, Pitman.

McCloskey M, Egeth H E (1983) 'Eyewitness identification: what can a psychologist tell a jury?', *American Psychologist*, May, 550-63.

McDougall W (1920) *The Group Mind: A Sketch of the Principles of Collective Psychology With Some Attempt to Apply Them to the Interpretation of National Life and Character*, Cambridge University Press.

McGinnes S (1991) *Neural Network Generation of Contour Maps*, Graduate diploma thesis, Department of Computer Science, Queensland University, Australia.

McGinnes S (1992) 'How objective is object-oriented analysis?', *Advanced Information Systems Engineering*, Proceedings CAiSE '92, Manchester, UK, Springer-Verlag.

McGinnes S (1993) 'A framework for the comparative evaluation of information systems methodologies', *Theory, Use and Integrative Aspects of Information Systems Methodologies*, Proceedings of the BCS/UKSS Joint Conference, Edinburgh, August 1993, British Computer Society.

McGinnes S (1994a) *A Framework for the Comparison of Object-Oriented Analysis Methods*, Master's Thesis, Faculty of Information Technology, Queensland University of Technology.

McGinnes S (1994b) 'Communication and collaboration: skills for the new IT professional', *Proceedings of the Second Annual Conference on the Teaching of Computing*, August, Dublin City University, Ireland.

McGinnes S (1994c) 'CASE support for collaborative modelling: re-engineering conceptual modelling techniques to exploit the potential of CASE tools', *Software Engineering Journal*, 9 (4), 183-189.

McGregor D R (1991) 'Reusability—the major promise and challenge of the object oriented approach', *IEE Colloquium on Applications and Experience of Object-Oriented Design*, 24th January.

McGregor J D (1990) 'Object-oriented design—introduction', *Communications of the ACM,* 33 (9), 38-39.

McKenna P J (1998) *Prospero and Caliban: Working Software from E-R Models*, MSc thesis, Department of Computer Science, Trinity College Dublin, Ireland.

McMahon L (1999) *3-D Model Viewer*, Undergraduate Project Thesis, Department of Computer Science, Trinity College Dublin, Ireland.

Medin D L (1989) 'Concepts and conceptual structure', *American Psychologist*, 44, 1469-1481.

Medin D L, Ross B H (1997) *Cognitive Psychology* (4th ed.), Harcourt Brace.

Meyer B (1988) *Object-Oriented Software Construction*, Prentice Hall.

Miele F (1979) 'Cultural bias in the WISC', *Intelligence*, 3 (April-June), 149-163.

Miller G A (1956) 'The magic number seven, plus or minus two: some limits on our capacity to process information', *Psychological Review*, 63, 81-97.

Miller G A (1975) *The Psychology of Communication,* Basic Books.

Miser H J, Quade E S (1985) *Handbook of Systems Analysis*, North-Holland.

Monteiro E, Hanseth O (1996) 'Social shaping of information infrastructure: on being specific about the technology', W J Orlikowski, G Walsham, M R Jones and J I DeGross (eds.) *Information Technology and Changes in Organizational Work*, Chapman and Hall, 325-343.

Montgomery S L (1994) *Object-Oriented Information Engineering: Analysis, Design, and Implementation,* Academic Press.

Mooney G (1998) Private communication.

Muller J M, Kuhn S (1993) 'Participatory design', *Communications of the ACM*, June 1993, 36 (4), 24-28.

Muller M J, Wildman D M, White E A (1993) '"Equal opportunity" participatory design using PICTIVE', *Communications of the ACM,* 36 (4), 64-66.

Mullin M (1990) *Rapid Prototyping for Object-Oriented Systems*, Addison-Wesley.

Mumford E (1981) 'Participative systems design: structure and method', *Systems, Objectives, Solutions*, 1 (1), 5-19.

Mumford E (1983) *Designing Human Systems*, Manchester Business School, UK.

Mumford E (1993) 'The ETHICS approach', *Communications of the ACM*, 36 (6),82.

Mumford E (1996) *Systems Design: Ethical Tools for Ethical Change*, Macmillan.

Mumford E, McDonald B W (1989) *XSEL's Progress - The Continuing Journey of an Expert System*, Wiley.

Mumford E, Weir M (1979) *Computer Systems in Work Design: The ETHICS Method*, Associated Business Press.

Munro M C, Davis G B (1977) 'Determining management information needs: a comparison of methods', *MIS Quarterly*, 1, 55-67.

Myers D G, Lamm H (1976) 'The group polarization phenomenon', *Psychological Bulletin* 83 (July), 602-27.

Mylopoulos J (1991) 'Representing knowledge about information systems', *Proceedings of the IJCAI workshop: Intelligent and Cooperative Information Systems: Bringing AI and Information Technologies Together*, August, Sydney.

Nandhakumar J (1997) 'Issues in participant observation: a study of executive information systems development', McKenzie G, Powell J, Usher R (eds.) *Understanding Social Research*, Falmer Press, 210-220.

Nandhakumar J, Avison D E (1999) 'The fiction of methodological development: a field study of information systems development', *Information Technology and People*, 12 (2), 1999.

Newsted P R, Chin W, Ngwenyama O, Lee A (1996) 'Resolved: surveys have outlived their usefulness in information systems research', *Proceedings of the International Conference on Information Systems, December 17, 1997*,Cleveland, Ohio.

Nguyen G T, Rieu D (1989) 'Schema evolution in object-oriented database systems', *Data and Knowledge Engineering*, (1989) 4, 43-67.

Ngwenyama O (1991) 'The critical social theory approach to information systems: problems and challenges', Nissen H E, Klein H K, Hirschheim R (eds.) *Information Systems Research: Contemporary Approaches and Emergent Traditions, Proceedings of the IFIP TC8/WG8.2 Working Conference on the Information Systems Research Arena of the '90s Challenges, perceptions and Alternative Approaches, Copenhagen, Denmark, 14-16 December, 1990*, North-Holland.

Nielsen J (1988) 'Evaluating the thinking aloud technique for use by computer scientists', *IFIP Working Group 8.1 International Workshop on Human Factors of Information Systems Analysis and Design (IFIP WHISAD'88)*, London, United Kingdom, July 28-29.

Nielsen J (1994) *Usability Engineering*, AP Professional.

Nierstrasz O (1989) 'A survey of object-oriented concepts', Kim W, Lochovsky H (eds.) *Object-Oriented Concepts, Databases and Applications*, ACM Press.

Nierstrasz O, Tsichritzis D (1989) 'Integrated office systems', Kim W, Lochovsky H (eds.) *Object-Oriented Concepts, Databases and Applications*, ACM Press.

Nijssen G M et al (1990) 'Entity-relationship data model considered harmful', Zunde P, Hocking D (eds.) *Proceedings of the 5th Symposium on Empirical Foundations of Information and Software Science*, Atlanta, Georgia, Plenum Press, 109-130.

Norman D A (1976) *Memory and Attention: An Introduction to Human Information Processing* (2nd ed.), Wiley.

Norman D A (1983) 'Some observations on mental models', Gentner D, Stevens A L (eds.) *Mental Models*, Lawrence Erlbaum, 7-14.

O'Donnell L (1997) *Annotated Video*, Undergraduate Project Thesis, Department of Computer Science, Trinity College Dublin.

Odell J J (1991) 'Object-oriented analysis', *Proceedings of the First IFIP Working Conference on the Object-Oriented Approach in Information Systems*, Quebec.

Odell J J, Fowler M (1997) *Advanced Object-Oriented Analysis and Design Using UML*, SIGS Books.

Ohnishi A (1994) 'A visual software requirements definition method', *Proceedings of the IEEE International Conference on Requirements Engineering*, 194-201.

Olle T W (1988) 'Business analysis and system design specifications for an inventory control and purchasing system', Olle T W, Verrijn-Stuart A A, Bhabuta L (eds.) *Computerised Assistance During the Information Systems Life Cycle*, North-Holland.

Olle T W (1988) 'System design specifications for a conference organisation system', Olle T W, Verrijn-Stuart A A, Bhabuta L (eds.) *Computerised Assistance During the Information Systems Life Cycle*, North-Holland.

Olle T W et al (1991) *Information Systems Methodologies: A Framework for Understanding* (2nd ed.), Addison Wesley.

Olle T W, Sol H G, Tully C J (eds.) (1983) *Information Systems Design Methodologies: A Feature Analysis*, North-Holland.

Olle T W, Sol H G, Verrijn-Stuart A A (eds.) (1982) *Information Systems Design Methodologies—A Comparative Review*, North-Holland.

Olle T W, Sol H G, Verrijn-Stuart A A (eds.) (1986) *Information Systems Design Methodologies: Improving the Practice*, North-Holland.

Opdahl (1994) 'A taxonomy for real-world modelling concepts', *Information Systems* 19 (3), 229-241, Pergamon.

Opdahl A L, Sindre G (1995) 'Representing real-world processes', *Proceedings of HICCS'28 minitrack on dynamic modelling,Hawaii/U.S.A,* January 1995.

Opdahl A L, Sindre G (1996) 'Real-world modelling', J G Williams (ed.) *Encyclopedia of Computer Science and Technology,* Marcel Dekker.

Orlikowski W J, Baroudi J J (1991) 'Studying information technology in organisations: research approaches and assumptions', *Information Systems Research*, 2,1-28.

Orpen C (1979) 'The effects of job enrichment on employee satisfaction, motivation, involvement, and performance: a field experiment', *Human Relations*, 32 (3), 189-217.

Osborn A F (1957) *Applied Imagination*, Scribner's.

Ould M A (1995) *Business Processes*, John Wiley.

Paivio A (1971) *Imagery and Verbal Processes*, Holt, Rinehart & Winston.

Paivio A (1986) *Mental Representation: A Dual-Coding Approach*, Oxford University Press.

Parnas D, Clements P (1986) 'A rational design process: how and why to fake it', *IEEE Transactions on Software Engineering*, Feb 1986.

Parnas R, Clemens P (1986) 'A rational design process: how and why to fake it', *IEEE Transactions on Software Engineering*, 12 (2).

Partridge C (1996) *Business Objects: Re-Engineering for Re-Use*, Butterworth-Heinemann.

Pasmore W A (1988) *Designing Effective Organisations: The Sociotechnical Systems Perspective*, Wiley.

Pinel J P (1993) *Biopsychology,* Allyn and Bacon.

Popkin (1995) *System Architect User Guide and Reference Manual,* Popkin Software and Systems Inc.

Powell P (ed.) (1998) Editorial, *Newsletter for the UK Academy for Information Systems*, March 1998 (Volume 4, Number 1).

Pylyshyn Z W (1991) 'Some remarks on the theory-practice gap', Carroll J M (ed.) *Designing Interaction: Psychology at the Human-Computer Interface*, Cambridge University Press.

Quatrani T (1998) *Visual Modelling With Rational Rose and UML,* Addison-Wesley.

Quinlan P T (1991) *Connectionism and Psychology: A Psychological Perspective on New Connectionist Research*, Harvester Wheatsheaf.

Reiss N (1985) *Speech act taxonomy as a tool for ethnographic description: an analysis based on videotapes of continuous behaviour in two New York households*, John Benjamins.

Remenyi D, Williams B (1996) 'The nature of research: qualitative or quantitative, narrative, or paradigmatic?', *Information Systems Journal*, 6 (2), 131-146.

Reubenstein H B, Waters R C (1991) 'The requirements apprentice: automated assistance for requirements acquisition', *IEEE Transactions on Software Engineering*, 3 (17), March 1991, 226-240.

Rine D (1991), 'The use of object-oriented construction to develop an expanded management information system.' *Cybernetic and Systems: an International Journal*, 22, 89-112.

Roberts S A, Gahegan M N, Hogg J, Hoyle B (1991) 'Application of object-oriented databases to geographic information systems', *Information and Software Technology*, 1 (33).

Robins K, Webster F (1989) *The Technical Fix: Education, Computers and Industry*, Macmillan.

Rochfeld A, Tardien H (1983) 'Merise: an information systems design and development methodology', *Information and Management*, 6, North Holland.

Rockart J F (1987) 'Chief executives define their own data needs', *Information Analysis: Selected Readings*, Robert Galliers (ed.) Addison Wesley.

Roethlisberger F J, Dickson W J (1939) *Management and the Worker,* Harvard University Press.

Rogers G (1988) 'Visual programming with objects and relations', *Proceedings of the 1988 IFIP Workshop on Visual Languages*, 38-44.

Rohlen T P (1975) 'The company work group', Vogel E (ed.) *Modern Japanese Organisation and Decision-Making*, Tuttle.

Rohr G (1988) 'Graphical user languages for querying information: where to look for criteria?', *Procedings of the 1988 IEEE Workshop on Visual Languages*, 21-28.

Rosch E (1975), 'Cognitive representations of semantic categories', *Journal of Experimental Psychology: General* 104: 192-253.

Rosson M B, Gold E (1989) 'Problem-solution mapping in object-oriented design', *Proceedings of the Conference on Object-Oriented Systems, Languages and Applications (OOPSLA '89)*, October 1-6 1989, 7-10.

Rosson M B, Sherman R A (1990) 'The cognitive consequences of object-oriented design', *Human-Computer Interaction*, (1990) 5, 345-379.

Ruhleder K, Jordan B (1997) 'Capturing complex, distributed activities: video-based interaction analysis as a component of workplace ethnography', Lee A S, Liebenau J, DeGross J I (eds.) *Information Systems and Qualitative Research: Proceedings of the IFIP TC8 WG8.2 International Conference on Information Systems and Qualitative Research, 31st May-3rd June 1997,* Philadelphia, Pennsylvania, USA, p.69, Chapman and Hall.

Rumbaugh J et al (1991) *Object-Oriented Modelling and Design*, Prentice Hall.

Rumelhart D E, McClelland J L (1986) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (Vols 1 and 2), MIT Press.

Sanford A J (1987) *The Mind of Man: Models of Human Understanding*, Harvester Wheatsheaf.

Schank R C (1982) *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*, Cambridge University Press.

Schank R C (1985) 'Reminding and memory organisation', A M Aitkenhead, J M Slack (eds.) *Issues in Cognitive Modelling: A Reader*, Lawrence Erlbaum.

Schank R C (1986) *Explanation Patterns: Understanding Mechanically and Creatively*, Lawrence Erlbaum.

Schank R C, Abelson R P (1977) *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge,* Lawerence Erlbaum.

Schneider G, Winters J P (1998) *Applying Use Cases: A Practical Guide,* Addison-Wesley.

Schneider W, Shiffrin R M (1977) 'Controlled and automatic information processing: I: Detection search and attention', *Psychological Review*, 84, 1-66.

Schön D A (1983) *The Reflective Practitioner: How Professionals Think in Action,* Harper Collins.

Schönpflug W (1986) 'The trade-off between internal and external memory storage', *Journal of Memory and Language* 1986, 25, 657-75.

Schwartzmann H (1993) *Ethnography in Organisations'*, Sage Publications.

Schweickert R, Boruff B (1986) 'Short-term memory capacity: magic number or magic spell?', *Journal of Experimental Psychology, Learning, Memory and Cognition*, 12, 419-25.

Scott Morton M S (1984) 'The state of the art of research', McFarlan F W (ed.) *The Information Systems Research Challenge*, Harvard Business School Press.

Seligmann P S, Sol, Wijers (1989) 'Analysing the structure of information systems methodologies: an alternative approach', *Proceedings of the 1st Dutch Conference on information systems*, Amsterdam, Neths, Nov 1-2.

Selker T, Koved L (1988) 'Elements of visual language', *Proceedings of the 1988 IFIP Workshop on Visual Languages*, 38-44.

Sernadas A, Fiadeiro J, Sernadas C, Ehrich H D (1989) 'The basic building blocks of information systems', E D Falkenberg, P Lindgreen (eds.) *Information Systems Concepts: An Indepth Analysis,* North-Holland, 225-246.

Sernadas C, Fiadero J (1991) 'Towards object-oriented conceptual modelling', *Data and Knowledge Engineering*, 6, 479-508.

Shaffer L H (1975) 'Multiple attention in continuous verbal tasks', P M Rabbitt and S Dornic (eds.) *Attention and Performance,* 5, Academic Press, 157-167.

Shastri L (1988) *Semantic Networks: An Evidential Formalization and its Connectionist Realization*, Pitman.

Shaw M E (1981) *Group Dynamics: The Psychology of Small Group Behaviour* (3rd ed.), McGraw-Hill.

Sheffield A D (1936) *Creative Discussion*, Association Press.

Shilling J J, Sweeney P F (1989) 'Three steps to views: extending the object-oriented paradigm', *Proceedings of the Conference on Object-Oriented Systems, Languages and Applications (OOPSLA '89)*, October 1-6 1989, 353-361.

Shlaer S, Mellor S (1988) *Object-Oriented System Analysis Modelling the World in Data*, Prentice Hall.

Shlaer S, Mellor S (1992) *Object-Oriented System Analysis Modelling the World in States*, Prentice Hall.

Shu N C (1989) 'Visual programming: perspectives and approaches', *IBM Systems Journal*, 28 (4), 525-547.

Shumate K (1991) 'Structured analysis and object-oriented design are compatible', *Ada Letters*, 4 (111), May/June 1991, 78-90.

Silverman D (1993) *Interpreting Qualitative Data*, Sage.

Simon H (1957) *Administrative Behavior,* Free Press.

Singer J L, Kolligan J (1987) 'Personality: developments in the study of private experience', *Annual Review of Psychology*, 38, 533-74.

Skinner B F (1950) 'Are theories of learning necessary?', *Psychological Review*, 57 (4), 193-216.

Smith E E, Medin D L (1981) *Categories and Concepts,* Harvard University Press.

Smith S M, Glenberg A, Bjork R A (1978) 'Environmental context and human memory', *Memory and Cognition*, 6, 342-53.

Snow C P (1959) *Two Cultures and the Scientific Revolution,* Cambridge University Press.

Solso R L (1998) *Cognitive Psychology* (3rd ed.), Allyn and Bacon.

Sowa J (1984) *Conceptual Structures: Information Processing in Mind and Machine*, Prentice Hall.

Sowa J F (1992) 'Logical structures in the lexicon', *Knowledge-Based Systems,* 5 (3), 173-182.

Spradley J P (1979) *The Ethnographic Interview*, Rinehart and Winston.

Sprague R H, Watson H J (eds.) (1996) *Decision Support for Management*, Prentice-Hall.

Stamper R (1987) 'How far towards a semantic normal form?', *Proceedings of the ASLIB Conference – Meaning: the Frontier of Informatics*, Cambridge, April 1987.

Stanley L, Wise S (1983) *Breaking Out: Feminist Consciousness and Feminist Research,* Routledge.

Stapleton J (1997) *DSDM: Dynamic Systems Development Method,* Addison Wesley.

Steiner I D (1972) *Group Process and Productivity*, Academic Press.

Stowell F A (1991) 'Towards client-led development of information systems', *Journal of Information Systems*, 1, 173-189.

Stowell F A, West D (1994) *Client-Led Design*, McGraw-Hill.

Straub D W (1991) 'Rigor in information systems research, Nissen H E, Klein H K, Hirschheim R (eds.) *Information Systems Research: Contemporary Approaches and Emergent Traditions, Proceedings of the IFIP TC8/WG8.2 Working Conference on the Information Systems Research Arena of the '90s Challenges, perceptions and Alternative Approaches, Copenhagen, Denmark, 14-16 December, 1990*, North-Holland.

Strauss A L (1987) *Qualitative Analysis for Social Scientists*, Cambridge University Press.

Sully P (1993) *Modelling the World With Objects*, Prentice Hall.

Sutcliffe A (1990) 'Human factors in information systems: a research agenda and some experience', Finkelstein A et al (eds.) *Human Factors in Information Systems Analysis and Design*, North-Holland.

Sutcliffe A (1991) 'Object-oriented systems development: survey of structured methods', *Information and Software Technology*, 33 (6).

Taggart W M, Tharp M O (1977) 'A survey of information requirements analysis techniques', *Computing Surveys*, 9 (4), Dec 1977, 273-290.

Tansel A U, Arkun M E, Ozsoyoglu G (1989) 'Time-by-example query language for historical databases', *IEEE Transactions on Software Engineering*, 4 (15), April 1989.

Tasker D (1991) 'Object lesson', *Computerworld*, April 22.

ter Hofstede A H M, Brinkkemper S (1989) *Conceptual Task Modelling*, Technical Report no 89-14, Department of Informatics, University of Nijmegen, Netherlands.

Thapar A, Greene R L (1994) 'The effects of level of processing on implicit and explicit tasks', *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 20, 671-679.

Thompson C P, Skowronski J J, Lee D J (1987) 'Reconstructing the date of a personal event', *Proceedings of the Second International Conference on Practical Aspects of Memory*, Swansea, UK 1987.

Tozer J (1992) *Object-Oriented Enterprise Modelling,* Prentice Hall.

Trauth E M, O'Connor B (1991) 'A study of the interaction between information technology and society: an illustration of combined qualitative research methods', Nissen H E, Klein H K, Hirschheim R (eds.) *Information Systems Research: Contemporary Approaches and*

Emergent Traditions, Proceedings of the IFIP TC8/WG8.2 Working Conference on the Information Systems Research Arena of the '90s Challenges, perceptions and Alternative Approaches, Copenhagen, Denmark, 14-16 December, 1990, North-Holland.

Treisman A, Gelade G (1980) 'A feature integration theory of attention', *Cognitive Psychology*, 18, 643-662.

Tsalgatidou A, Loucopoulos P (1990) 'An object-oriented rule-based approach to the dynamic modelling of information systems', *Dynamic Modelling of Information Systems*, Sol H G, van Hee K M (eds.) North-Holland 1991.

Turner J A (1991) 'Relevance versus rigor in information systems research: an issue of quality', in H-E Nissen, H K Klein and R Hirschheim (eds.) *Information Systems Research: Contemporary Approaches and Emergent Traditions*, North-Holland.

Urquhart C (1997) 'Exploring analyst-client communication: using grounded theory techniques to investigate interaction in informal requirements gathering', Lee A S, Liebenau J, DeGross J I (eds.) *Information Systems and Qualitative Research: Proceedings of the IFIP TC8 WG8.2 International Conference on Information Systems and Qualitative Research, 31st May-3rd June 1997,* Philadelphia, Pennsylvania, USA, Chapman and Hall.

Van assche F et al (eds.) (1991) *Object-Oriented Approach in Information Systems*, North-Holland.

Van Assche F, Layzell P, Loucopoulos P, Speltincx G (1988) 'Information systems development: a rule-based approach', *Journal of Knowledge-Based Systems*, September 1988, 227-234.

van Reijswoud V E, Mulder J B F (1997) 'Speech act based modelling of workflow management systems: a case study', *Proceedings of the Second International Workshop on Communication Modelling (LAP'97),* Veldhoven.

van Reijswoud V E, Mulder J B F (1998) 'Speech act based communication and information modelling with DEMO', *Australian Journal of Information Systems*, Vol 6, No 1, pp.89-102.

Varney G H (1989) *Building Productive Teams: An Action Guide and Resource Book*, Jossey-Bass.

Vennix J A M (1996) *Group Model Building: Facilitating Team Learning Using System Dynamics,* John Wiley.

Verbraeck A, Wierda F W (1990) 'Interactive modelling for information systems design: the mosaic tool', Sol H G, van Hee K M (eds.) *Dynamic Modelling of Information Systems*, North-Holland.

Veryard R A (1992) *Information Modelling: Practical Guidance*, Prentice-Hall.

Vitalari N P (1984) 'Critical assessment of structured analysis methods: a psychological perspective', *Beyond Productivity: Information Systems for Organisational Effectiveness*, Bemelmans T H (ed.) North-Holland 1984.

Vonk R (1990) *Prototyping: The Effective Use of CASE Technology,* Prentice-Hall.

Waddington D (1994) 'Participant observation', Cassell C, Symon G (eds.) *Qualitative Methods in Organisation Research*, Sage Publications.

Walsham G (1993) *Interpreting Information Systems in Organizations*, Wiley.

Walsham G (1996) 'Ethical theory, codes of ethics and information systems practice', *Information Systems Journal*, 6, 69-81.

Ward P (1986) 'The transformation schema: an extension of the data flow diagram to represent control and timing', *IEEE Transactions in Software Engineering*, 12 (2) 198-210.

Wasserman A I, Pircher P A, Muller R J (1990) 'The object-oriented structured design notation for software design representation', *Computer*, 23 (3), 50-63.

Weiss S, Page-Jones M (1991) *Synthesis: An Object-Oriented Analysis and Design Method*, Macmillan.

Weldon E, Gargano G M (1988) 'Cognitive loafing: The effects of accountability and shared responsibility on cognitive effort', *Personality and Social Psychology Bulletin*, 14, 159-71.

West M A (ed.) (1996) *Handbook of Work Group Psychology*, John Wiley.

Wilson B (1984) *System Concepts: Methodologies and Applications*, John Wiley.

Winograd T, Flores F (1986) *Understanding Computers and Cognition: A New Foundation for Design*, Ablex.

Wirfs-Brock R F, Johnson R E (1990) 'Surveying current research in object-oriented design', *Communications of the ACM*, 33 (9), 104-124.

Wirfs-Brock R J, Wilkerson B, Wiener L (1990) *Designing Object-Oriented Software*, Prentice-Hall.

Wood J, Silver D (1995) *Joint Applications Design: How to Design Quality Systems in 40% Less Time* (2nd ed.), Wiley.

Woodcock J C P, Loomes M (1988) *Mathematics for Software Engineering,* Pitman.

Wood-Harper A T, Antill L, Avison D E (1985) *Information Systems Definition: The Multiview Approach*, Blackwell.

Wood-Harper T (1985) 'Research methods in information systems: using action research', E Mumford et al (eds.) *Research Methods for Information Systems*, North-Holland 1985.

Yadav S B (1983) 'Determining an organisation's information requirements: a state of the art survey', *Data Base*, Spring 1983, 3-20.

Yin R K (1989) *Case Study Research: Design and Methods,* Sage.

Yourdon E (1977) *Structured Walkthroughs*, Yourdon Press.

Yourdon E (1989) *Modern Structured Analysis*, Prentice-Hall.

Yourdon E, Whitehead K, Thurmann J, Nevermann P, Oppel K (1995) *Mainstream Objects*, Prentice Hall.

Zachman J A (1982) 'BSP and BICS: a comparison', *IBM Systems Journal*, 21 (1).

Zachman J A (1982) 'Business Systems Planning and Business Information Control Study: a comparison', *IBM Systems Journal*, 21 (1), 31-53.

Zander A (1989) *Making Groups Effective*, Jossey-Bass.