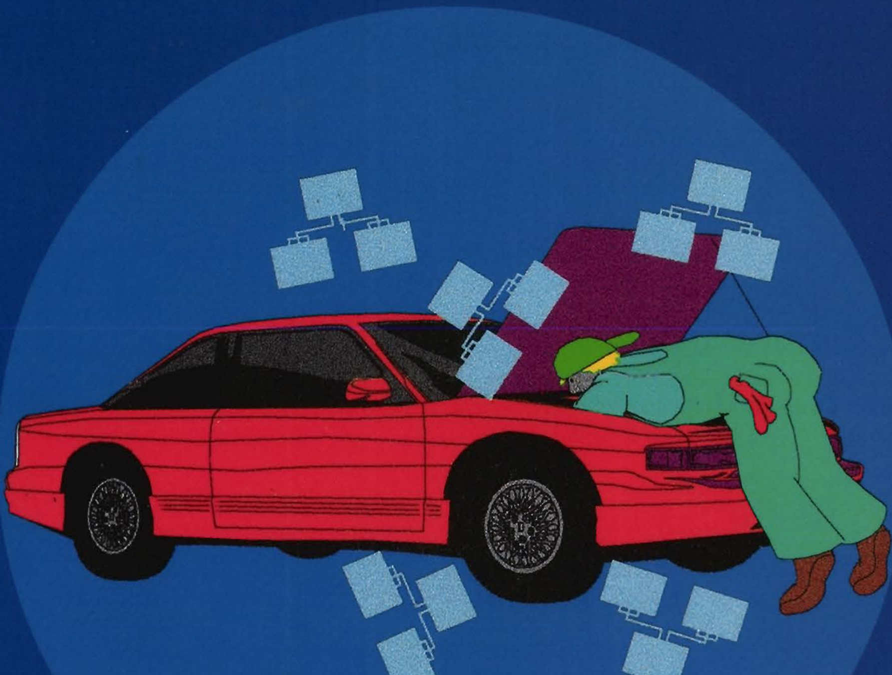
Assessment of business information systems by data structures

DOI:
10.6100/IR465328

Document status and date:
Published: 01/01/1996

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Assessment of Business Information Systems by Data Structures



Jos C.J. de Heij

Kluwer Bedrijfswetenschappen

Coopers & Lybrand

Assessment of
Business Information Systems
by Data Structures

Assessment of Business Information Systems by Data Structures

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de Rector Magnificus, prof.dr. M. Rem, voor een commissie aangewezen door het College van Dekanen in het openbaar te verdedigen op vrijdag 13 september 1996 om 16.00 uur.

door

Johannes Cornelis Jozef de Heij

geboren te 's-Gravenhage

Dit proefschrift is goedgekeurd door de promotoren:-

prof.dr.ir. J.C. Wortmann
en
prof.dr.ir. J.W.M. Bertrand

Preface

In this preface I would like to take the opportunity to thank everybody for their contribution to my doctoral research and to the realisation of this sizeable work. Unfortunately, it is impossible to mention everybody personally. However, this does not mean that their role is less significant, quite the contrary. An unexpected by-product of this magnum opus was a number of new friends. There are nevertheless a few people I would like to give a special mention.

First, I would like to thank my doctoral thesis supervisors Hans Wortmann and Will Betrand. Hans in particular motivated me again and again from my graduation (on the same subject) onwards, not only lending support for the contents but also personally. When I reached a stage at which I wanted to quit, Hans and Jan Moggré (my mentor at Coopers & Lybrand) convinced me that I really had to finish the job.

Second, I received much support from various (ex-)colleagues at Coopers & Lybrand for which I am most appreciative. Besides their personal interest, I was able to reflect the interim results against the ideas of many colleagues with their positive criticism. Both René Caubo and Paul Frenay had a major influence on the resulting contents. René originally helped me as a graduate with a properly founded version of the reference model. The co-operation was so good that we have become colleagues in the mean time. So now and then Paul was one of the few people who really believed in this controversial approach, especially when the criticism became visible in the public press. Thanks to his effort in part this criticism has turned into understanding and many positive reactions have been received.

Besides this encouragement, this research would never have come about without the help of various software suppliers. I would particularly like to thank the following companies for their efforts: Baan, InfoStore, JBA-Ratioplan, Marcam, Raet, SAP, SSA, and TXbase. They not only provided me with much, often confidential, information, but also a number of their employees were willing to spend their valuable time on answering my often difficult questions.

Naturally, I had a lot of support and understanding from my family, and my parents in particular. They have always stimulated me from the primary school on, and despite a laborious course of study this has finally led to this unexpected result.

Last but not least, I would like to extend my thanks to Lidy. She was (and is) my most important motivator, but also my help and stay, and without her this dissertation would not exist. For an outsider it may seem a miracle that our relationship did not suffer from the extreme pressure of Ph.D. research. On the contrary, despite all efforts and pressure we have weathered the storm and come out stronger. Having won this, I am positive we will live to see many more special occasions.

Jos C.J. de Heij
July 1996, Moerkapelle

ISBN 90-267-2475-6
1e druk 1996

Table of contents

Part 3: Test and Evaluation

Appendixes

Figures

Tables

Summary

1 Currently, the trend in business information systems (BIS) is moving towards acquisition of standard packages instead of companies developing their own information systems. In a growing number of industries, standard packages offer a good alternative. This development started in particular around systems for manufacturing control (the MRP packages). On the one hand, these packages are developing towards systems for controlling the total logistic chain, on the other, they tend to support *all* company functions for example, including financial and personnel management.

2 The current trend towards standard BIS is influenced by the rapid development of technology. Graphic user interfaces have become an important selection criterion. In particular, object-oriented technology might in the future enable combining separate object classes from different suppliers into one integrated BIS for example, a purchasing object class of one supplier combined with an inventory object class of another supplier.

The right choice of a standard business information system (BIS)

3 The BIS is a critical success factor for a company, success or failure of the implementation influence the position of the company on the market and its results. However, first of all a choice has to be made between a broad range of existing BIS. Taking into account the enormous diversity of companies and their features, it is clear that the right choice is essential.

4 Based on experience, a generally accepted approach to the selection of complex systems has developed. The 'best practice' approach contains a number of stages and activities, of which the most important are:-

(a) *the preliminary selection*, the preparation of a long list containing less than ten candidates, based on a very limited effort;

(b) *the detailed selection*, the limitation of the long list to a short list of not more than three BIS. This selection is carried out based on a limited number of essential assessment criteria (knock out criteria);

(c) *the assessment*, the actual assessment of the short list candidates with the aid of a company specific case during a demonstration;

(d) *the completion*, usually focused on entering into a contract and preparing the implementation.

5 An important selection aspect, though not the only one, is assessing and comparing the functional features of the candidate systems. In the approach described above, this is carried out based on user requirements which are then checked using a questionnaire and in demonstrations of the company case.

6 Preparing and assessing this questionnaire is quite a task. Suppliers often interpret the questions in different (sometimes creative) ways. Moreover, often the questionnaire demands a specific solution in the BIS while different options - which might not yet have been thought of - are also possible and sometimes better. On the other hand, a supplier confirming that he can offer a solution to a defined problem, does not guarantee any quality.

> Example: „*Can materials be purchased directly on sales order?*"; a „*Yes*" of the software supplier does not say anything about the way this is supported, the number of transactions and simplicity, and under which functional conditions.

7 An assessment based on a company case is certainly not watertight. The preparation of a good test case including all specific company and business process characteristics is a tall order. Whilst one or two days of demonstrating such a complex case to the future users is possible, it does not lead to an objective assessment by these users. There are other aspects that play a more important role, such as the commercial skills of the demonstrator, the appearance of the system and the extent to which the system links up to the existing (old) frame of reference of the assessor.

8 In view of the complexity and range of the BIS concerned, often many weeks sometimes months are needed in order to assess the system in an objective manner. This in fact is caused by the 'black box' approach for assessing systems as described above. This approach does not make use of any knowledge of the internal structure of systems.

Purpose of the study

9 This study originates from the idea that assessing the data structure of BIS will add important value to the quality of selections. Studying the core of a BIS, its data structure, can be compared to assessing the technology of a car by looking under the bonnet. *The study is therefore geared to the possibilities as well as the limitations of using data structures for the assessment of business information systems (BIS).*

The method for assessing data structures

10 For the benefit of this study I developed a method to compare the data structures of different BIS in a uniform manner. The core of this method is the reference data model (RDM). Instead of comparing the very diverse data structures of different BIS with each other, the RDM-method projects all these data structures onto the RDM. Based on these projections one can reach conclusions on the availability of functionality in the BIS concerned, the nature of the solution the system offers and its implementation. Furthermore, the projections can directly be compared to each other, enabling the assessor to concentrate on the differences.

11 I have developed a useful RDM for a number of function areas, for '*purchasing*' and '*state-independent manufacturing data*' (for example, bills of material, routings and recipes). A substantial part of this study consists of the elaboration of this RDM. By way of illustration, it describes different (partial) projections of existing BIS on this RDM. The RDM was primarily developed by combining a large number of data structures of existing BIS, including Prism, Ratio, SAP-R/3, Triton and TXBase. The result is an accumulation of knowledge and experience currently present at different independent suppliers in the functional areas studied.

12 As the RDM has a number of specific objectives, it fundamentally differs from the data structure of an information system to be developed, namely:-

(a) *its capabilities of definition*, which should result in uniform terminology;

(b) *its capabilities of projection*, enabling the projection of totally different data structures. *This could lead to the RDM offering alternative solutions which normally would not be combined in one logical data structure;*

(c) *its capabilities of distinction*, keeping important differences of different data structures visible after projection. In many cases this means that the data structures in the RDM are relatively detailed (or specialised);

(d) *its capabilities of creating insight*, keeping the philosophy of the original BIS visible.

13 A fundamental observation of this study is that the RDM developed in manner described above is not - and can never be - the data structure of a new (comprehensive) BIS, and must therefore not be confused with a universal data model (UDM). Typical differences between RDM and UDM are:-

(a) UDM chooses from alternatives, while the RDM combines alternatives;

> *By combining data structures the RDM will automatically contain different alternatives for one functional area, for example, a production model as well as the traditional routings and bills of material. In a UDM a fundamental choice must be made for one specific alternative depending on the target group.*

(b) UDM avoids redundancy, while the RDM makes use of redundancy;

> *Redundant data (or relationships) in BIS are only allowed as an exception. Sometimes manufacturing operations are defined first and the material requirements will be linked in later, alternatively, the bill of material is defined first and is later expanded with the operations. This combination of possibilities leads to a number of redundant relationships between groups of data. In the RDM this sort of redundancy is fundamental due to the complete combination of data structures, also where in a real information system this is of no value. An information system based on the RDM would result in many unnecessary transactions and extra data entry.*

(c) UDM is logically consistent, while the RDM does not have to be and because of the effects of (a) and (b) probably never will be.

14 Due to the combination of a wide range of BIS data structures of a very wide range, the RDM is not limited to one industry sector, one logistic controlling structure or type of manufacturing process. Therefore, the RDM is valuable for comparing different situations or typologies, but will not result in a standard industry model for any one sector.

Testing data structures as assessment tool

15 For the purpose of this study the quality of the assessment based on data structures was tested. This was checked by comparing the assessment with the suppliers answers to the conventional questionnaire.

16 I used an existing questionnaire prepared for a survey of software packages for manufacturing control. This publication, widely known on the Dutch market, includes the answers of all suppliers to an extensive list of questions regarding the functionality of the BIS they supplied. After the survey, independent consultants assessed the answers of the suppliers.

17 An assessment of these questions was also carried out based on the data structure for four different BIS. **In a significant number of cases this assessment differs** from the answers the suppliers gave. In order to verify this, the differences were discussed with the suppliers concerned. In most cases the supplier confirmed that the conclusion based on the data structure was correct, which, by the way, does not automatically mean that in all these cases the questionnaire answers given, were wrong. The main causes for these differences were:-

(a) the supplier interprets the question in a different manner than the actual intention of the question (many questions can be creatively explained in different ways);
(b) special 'work-arounds' (or 'tricks') that are only possible in the software by coincidence, lead to functionality that seems to be what is required. To which extent this functionality suffices differs per situation;
(c) differences in versions. The suppliers answer as well as the conclusion based on the data structure are essentially correct. The fact that the data structure model notices these differences confirms its assessment capabilities;
(d) omissions in the documentation in a number of assessed BIS.

18 In order to link the conclusions to the type of questions, the questions were classified. The main division is made between data-oriented and process-oriented questions. The data-oriented questions can directly be translated into single, multiple or complex data structures. The process-oriented questions are focused on supporting a business process or an activity in a process, arranging the activities in a process or the behaviour of these activities.

19 After the questionnaire concerned was classified, it became clear that the percentage of data-oriented questions versus process-oriented questions was 83% and 17% respectively. Thus it could be concluded that experts prepare the questionnaires specifically focused on data structures, although this was not an explicit part of the study. I have not studied to which extent this assumption applies to every questionnaire, nor has a statement been made concerning good and bad questions.

20 Quantitative test results are as follows:-

(a) the number of differences found where the data structure reached the correct conclusion is almost 14% of the data-oriented questions;

(b) the number of questions where the data structures could give no statement at all is about 17%;

(c) the number of data-oriented questions on which the data structures could give no statement is about 7%;

(d) in two cases the data structure implied more functionality than was actually implemented.

21 When analysing the questions for which no data structures statements can be made, two categories can be distinguished.

22 Firstly, the functionality that can be translated into single data structure characteristics, namely to individual attribute types, can often (but not always) be made acceptable with the aid of the detailed data dictionary. However, the *presence* of certain attribute types may not automatically be connected to the functionality desired, it is merely an indication. The *absence* forms a steady basis for a discussion with the supplier on how the functionality could be realised.

23 Secondly, for two types of process-oriented questions it is usually impossible to make statements based on the data structure (as is to be expected). This mainly concerns questions regarding the *behaviour of individual process steps* and the *sequencing of steps in processes.*

24 The quantitative results (see paragraph 20) will depend on the specific questionnaire used for the assessment. This questionnaire together with the verified answers give a clear indication of the strong and weak points of using data structures. The relatively high percentage of differences noted, clearly confirms this. If the test would have taken place based on a 'perfectly answered questionnaire', the conclusions drawn from the data structure would not differ. However, this only means that this check cannot confirm the usefulness of data structures, after all, the comparison is made to a theoretical, ideal situation.

Evaluation and conclusions

25 Data structures are a necessary condition, but not a sufficient condition for the presence of functionality. This implies that the only *absence* of the data structure leads to a definite conclusion.

26 Therefore, when trying to prove the *presence* of functionality, assessing the data structure will only indirectly say something about the **possible** functionality present. A data structure without software will not offer the functionality desired. However, supported by my testing experiences, this does not seem to lead to an insurmountable problem when assessing standard BIS. If the data structure is present in the form of (complex) entity types and relationship types, usually the functionality is also present. Designing such a complex data structure is one of the most important development activities. Taking into account this effort, it would be a lost opportunity to ignore the other development activities when developing a BIS. On the other hand, if only the fundament in the form of a data structure is present, it is relatively inexpensive to add the desired functionality.

27 However, the assumption with regard to the presence of functionality is not valid for (simple) individual attribute types. One should be more cautious drawing conclusions in this case. One will often find attribute types containing the description 'for future use'.

28 The test confirms that data structures give insight into the functional possibilities of BIS. This insight is proven by the following special features compared to for instance, a functional questionnaire:-

(a) the data structure shows the context in which the functionality is supported. It in particular shows which combinations of functionality are possible, or mutually exclusive;
(b) the data structure makes the BIS philosophy visible i.e., the terminology used, the aggregation levels of entity types, the choices made and the supported (manufacturing, purchasing and business) structure;
(c) a data structure has a number of abstraction and description levels, making it possible to zoom in and out of this description form as desired;
(d) the data structure contains much more information than could be thought of or defined beforehand. The answers to a questionnaire and the information gathered during a demonstration are limited to the aspects that seemed relevant at the time; whereas a data structure can subsequently offer new relevant information. After all, a data structure is a complete description of the data aspect.

29 The most important limitations when using data structures and the RDM-method in particular, are:-

(a) the expertise required of the assessor;
(b) the investment necessary to develop a good RDM and to initially project the data structures of all important BIS (this limitation specifically applies to the RDM-method and not to using data structures in general);
(c) the incompleteness of the assessment. There is a limited, although known, category of requirements that can not be predicted.

30 Moreover, one should realise that data structures describe solutions (choices and implementations) and are not meant to describe a problem (which functionality is necessary). Several data structures can offer a solution to one problem i.e., data structures are not suitable for defining a general need or desired structure of a standard BIS. The 'problem' is not a data structure, the data structure is the answer to a question regarding required functionality.

31 During the detailed selection, when a short list of BIS is prepared, the data structures could play a similar role as a database with functional features of all BIS. One of the advantages compared to a database is the fact that there is much more information available in data structures than can be defined beforehand. When assessing the short list of candidate BIS, the data structure of these BIS will provide information on critical points which definitely need testing during the demonstration.

32 The appropriate expertise is of vital importance in all areas in order to handle and interpret the data structures correctly.

Added value in the selection of BIS

33 The correct use of data structures as an assessment tool can have the following added value in the process of selecting a standard BIS:-

(a) *systematic recognition of critical points* in standard BIS (data structure);

(b) *recognising large modifications necessary at an early stage (including an estimate of the effort necessary),* which result from the test of critical points in the data structure;

(c) *minimising the chance that* ***unexpected*** *fundamental modifications are necessary,* the data structure has been verified and the fundamental nature of the modifications usually concern changing the (data) structure;

(d) *guaranteed appropriate (=correct) choice,* in the form of an appropriate data structure, and therefore the basic functionality.

Part 1

Introduction and Problem Definition

Introduction

1 This book is the result of research into the role that data structures may play in the assessment of Business Information Systems (BIS). This group of information systems encompasses the enterprise-wide and integrated information systems that occur in the sector of industry. Nowadays, the term ERP-systems (Enterprise Resource Planning) is often used for a type of system that considerably overlaps the above definition. However, considering the fact that the term ERP is widely used, I have refrained from using it in this context and chosen to use the more general and less loaded term BIS.

2 This research was aimed at standard BIS and not at information systems that are specifically developed by (or under the auspices of) one particular enterprise. Such custom-made systems were not studied in this framework. That means that any reference is made to BIS in this book, it principally refers to standard software systems.

3 The aspect of assessment plays an important role in standard software, to help make the proper choice from, among other things, the varied market. Until now, no structural use has been made of the BIS data structure. Nevertheless, the data structure offers interesting opportunities. Hence, an investigation was started to find out the possibilities and limitations of data structures as an assessment instrument.

4 The first part of this book gives an introduction to the subject of BIS. The introduction is not necessary for the definition of the problem under investigation or the research, but it does provide an overview of the historical developments, the development of the underlying technology, its influence on the present generation of BIS, and the practice of selecting BIS. This introduction encompasses chapters 1 and 2.

5 In chapter 3 the actual definition of the problem under investigation is stated based on analysis of experiences with the assessment of BIS. This analysis results in the research objective and approach stated at the end of this chapter.

6 In the second part of the book a method is developed for the comparison of data structures. The method is based on a reference data model: the RDM. In chapter 4 the design criteria for the RDM are deduced and the characteristics are further explained.

7 Chapters 5 and 6 give an extensive description of a RDM for the functional areas 'Purchasing' and 'State-independent Manufacturing Data' (all static data that describe products, product structures, resources, production processes and their control). For both areas this RDM basically encompasses all existing knowledge from a multitude of BIS of many different sources, supplemented by the knowledge and experiences gained from literature. Though the RDM is incomplete by definition and always will be, it has a major stand-alone value because of its systematics, structure, and level of detail. Take for example its value for training and knowledge transfer, or the design of business information systems.

8 The third part of this book is aimed at the actual testing and evaluation of the use of data structures as an assessment instrument. In chapter 7 the testing procedure used is described. Here, use is made of an existing questionnaire concerning the functional characteristics of BIS. This questionnaire was also filled in on the basis of the data structure, which leads to a comparison of the answers in this chapter.

9 In chapter 8 the evaluation is carried out based on the differences found in the comparison and the experiences in the design process of the RDM and the RDM-method. The conclusions lead particularly to proposals for practical use in software selection.

10 In chapter 9 possible consequences of the (rapid) technological development in this data-oriented approach are discussed.

11 Finally, in chapter 10 a number of proposals are made for supplementary research, as well as the practical application.

12 Appendix A gives a brief explanation of the terminology used in the area of the representation of data structures. In Appendix B the schematic technique itself is described. Neither appendix is intended as a means to learn the design and interpretation of data structures. For this the reader is referred to the literature in the area of data modelling.

13 Appendix C contains the questionnaire used. In addition, individual questions have been explained further and commented upon where this was desirable.

14 Appendix D is the Dutch summary of this book, which is identical to the English summary that precedes this introduction. Appendix E is an overview of all literature references made in this book. Finally, Appendix F contains a short résumé of the researcher and writer of this book.

1 State of the art in Business Information Systems and the reasons for studying data structures

1001 Business Information Systems (BIS) have an increasingly important role to play within organisations and also between organisations. Integrated software supporting such business information systems have already undergone extensive standardisation, particularly in the trade and industry sector. The selection of standard software is no easy task because of the impact the software has on the organisation and the costs involved. The need to assess standard software on the basis of its data structure has arisen from the limitations of the present software assessment methods.

1002 This chapter is mostly an introduction to the history and development of business information systems in the industry sector. The introduction is merely intended to provide the reader with some background information with regard to this fascinating subject. Therefore this chapter is not a fundamental scientific essay. The scientific problem definition and the research approach on using data structures for software assessment is described in chapter 3.

1003 As an introduction, the historical developments in the area of standard logistics software are outlined from paragraph 1100 onwards. The developments have led to the present considerable variety of systems. In order to make a choice some practical software selection examples are given.

1004 The research is aimed at the assessment of BIS which is one of the most important activities in the process of software selection. Within the context of BIS selection, an overview is given of the pitfalls and risks involved from paragraph 1200 onwards.

1005 By studying the data structures of BIS, a number of these pitfalls may possibly be avoided. From paragraph 1300 onwards, some of the reasons for studying BIS data structures are given and thus a structured approach to improving the assessment of BIS.

1006 Finally, from paragraph 1400 onwards important technological developments in relation to BIS are considered. The rapid technological developments have proved to be of major influence on the standardisation, continuity of and communication between BIS. There is left no choice but to involve these developments in any approach or method for software assessment.

From in-house systems development to standard Enterprise Resource Planning systems

1100 The present generation of Enterprise Resource Planning[1] (ERP) systems is the result of a development that started about forty years ago. Here, the rough steps within this development will be dealt with, as well as the key players at that time. Next, the present supply of BIS is put into perspective and a brief look will be taken at the most important aspects in considering alternative systems. Finally, some considerations are given that play a role in the selection of such complex systems.

---

1 The term Enterprise Resource Planning (ERP) is used in many contexts. Since this may cause confusion, the more generalised term Business Information System (BIS) has been chosen instead.

History in the development of manufacturing software

1101 The development of logistics software actually started with simple stock management systems. On the basis of stock controlling formula's like those of Camp, the first software packages for stock management were developed in the late fifties and early sixties. One of the first of these 'software packages' were RAMAC (Random Access Method for Accounting and Control) dating from 1957 and IMPACT (Inventory management Program And Control Techniques) developed by IBM in 1960. These packages were intended to manage large numbers of items, make forecasts on future sales, calculate safety stock and determine the size of purchase orders, and of course everything was focused on stock-oriented items.

1102 With the growing capabilities of computers, the automation of extensive calculations for the full bill of material explosion became possible. This development was made possible through the introduction of direct access disk drives. These early manufacturing management packages were called BOMP (Bill Of Material Processor). Generally, they were copies of the calculation methods done by hand at the time, usually based on monthly buckets (time periods in which a requirement is measured). During the course of the years they developed into complete MRP (Material Requirements Planning) systems. In the early seventies, the MRP philosophy that had been developed was given theoretical footing by Orlicky, Plossi and Wight [ORL72].

1103 Some important functional issues in the MRP packages at that time were the bucketless systems (capability of making accurate plans on an daily basis), pegging (tracing the sources of requirements) and net change (minimise the calculations by only recalculating changes). Generally these systems were intended for stock controlled manufacturing.

1104 Due to the needs of make-to-order manufacturers, parallel developments were taking place in the area of capacity planning and shop floor control. One particular example is CLASS (Capacity Loading And Scheduling System) by IBM. However, its integration with MRP packages, like IBM's COPICS, never really took off.

1105 Instead, the suppliers of MRP packages developed supplementary modules in the seventies for capacity requirements planning (CRP) and rough cut capacity planning (RCCP). For this the standard APICS concept was adopted, which resulted in the so-called MRP-II packages (Manufacturing Resource Planning).

1106 In the early seventies the number of standard software packages was very limited, approximately 40 in 1974. Many (large) organisations developed their own information systems in-house, by themselves. The development of standard packages was strongly pushed by the hardware suppliers. Examples of well-known packages are UNIS written by Sperry, PCS written by Burroughs and IPICS (Initial Production Information and Control System) written by IBM for the S/3.

1107 In the second half of the seventies on-line/real time features were becoming increasingly important and various mini computers were starting to hold their own: IBM S/34, S/36, S/38 (the predecessors of the contemporary AS/400), but also vendors like HP (HP-250, HP-1000, HP-3000, and HP-9000), Wang, DEC and Prime. The on-line integration between the various modules such as sales, manufacturing, purchasing, and finance were becoming more important. Well known examples of packages available towards the end of the seventies are MAPICS by IBM, MM/3000 by HP, MAC-PAC by Anderson, and MANMAN by ASK.

1108 From that moment on integration has remained an important issue and shifted to new functional areas like warehouse management, automated storage and retrieval, process control and labour reporting (integration still is an important issue today, see paragraph 1111). In the mid-eighties standard packages developed broader functionality and a number of suppliers started to specialise. Packages emerged that were specifically intended for the semi-process industry and for enterprises working on a project basis.

1109 By the early nineties, standard software has became widely accepted and in-house systems development became no longer viable. Nevertheless, the packages on the market are far from having completed their development. Examples of important selection criteria from a few years ago include: use-friendliness, international/multi-currency, two-level master scheduling, integrated multi-plant, variant bills of material and product configurators, (de)centralised organisation structures, by/co-products, line scheduling, blanket purchase orders, sales/purchase quotations. Most of these criteria are still discriminating when selecting software packages.

Now we want Enterprise Resource Planning software

1110 After MRP-I and MRP-II the term ERP (Enterprise Resource Planning) has become fashionable. What exactly is meant is not always clear; often vendors wish to distinguish themselves from the competition by using such terms. The Gartner Group made an attempt to make ERP more concrete: „whereas MRP is reactive and focuses on transactions and reporting, ERP is pro-active and provides decision support as an integral part of the entire information system" [GAR94.1/2].

1111 According to the Gartner Group, typical technological characteristics of ERP-systems are: relational, open, GUI (graphical user interface), and portability to different hardware platforms. Functional characteristics include:-

(a) the integration of manufacturing and logistics with other functional areas such as finance, sales & marketing, distribution, transportation, maintenance, field-service, and human resources;
(b) instead of a hierarchical and centralist structure, support for decentralised structures (multi-site databases), in which the primary business functions can be performed both in a centralised and in a decentralised way;
(c) the support of flow-based manufacturing, manufacturing without work orders even in job-shop like environments;
(d) the support of hybrid manufacturing environments (mixed mode): combined control with and without work orders, the use of bills of material and routings, as well as the 'production model';
(e) above the level of the main production plan, there is a possibility to plan in terms of financial units;
(f) the supply chain integration of customers (and their customers) and suppliers: supply chain management with EDI-tools, links to distribution depots, public warehouses, and large retailers;
(g) the presence of product configurators, both in the trend towards customer order controlled manufacturing, also in the (semi-)process industry.

1112 Functionally ERP can be viewed as an evolutionary step after MRP-II. However, in terms of technology and architecture ERP should be considered as a revolution.

1113 Yet in daily practice there is regular confusion about whether a software package is an ERP-system or not. Within the framework of this investigation into data structures, the term ERP is therefore avoided and we generally speak of Business Information Systems (BIS) instead. The figure below shows an overview of the most important players in this area.

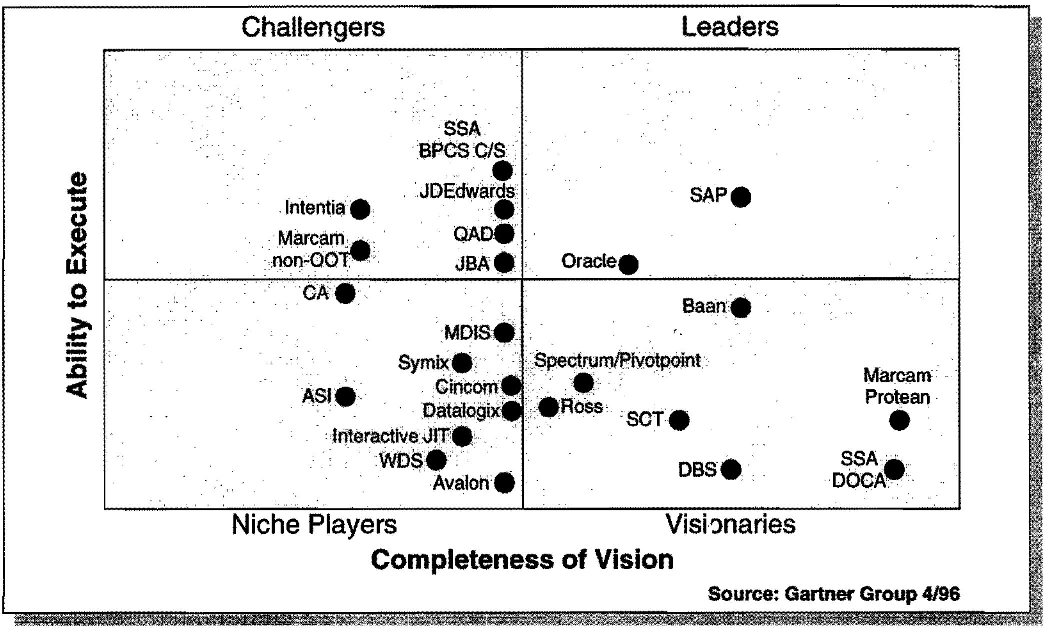


Figure 1.1 - Positioning of today's Business Information Systems

1114 The positions shown above are changing rapidly. Experts expect a major shake-out of the traditional systems to take place. In order to survive, Business Information Systems will have to adapt to the latest trends in industrial enterprises, such as global / multi-site enterprises, the integration of distribution and manufacturing within one enterprise, the hybrid manufacturing situations called upon by diversification, the (continuous) business (process) re-engineering, and a second generation of users of logistics software.

1115 The Gartner Group distinguishes six dimensions that are measured on a scale for the positioning of (ERP) systems with respect to each other:-

(a) *functionality.* Naturally, this is always an important positioning and selection criterion. Demands are changing rapidly now that systems are becoming broader and offer more and more detailed functionality. Reference was made to new important functional area's;

(b) *service/support.* The preference is one world-wide main supplier. In practice, however, an increasing number of suppliers are co-operating with business partners like system integrators, third parties, distributors, independent consultants, and hardware vendors. This does not actually benefit support in practice. Here, too, the quality of maintenance plays a role, which is often seen as free money and a cash cow;

(c) *vendor profile.* Among other things, this is assessed on the basis of turnover, profit, and research and development. The financial means to continue to invest in new technologies determine the continuity of the system and the supplier;

(d) *user references.* An attempt to ascertain to what extent a system has proved itself in practice. The number of installations is measured together with customer satisfaction, as well as information from user groups;

(e) *technology.* It will be of crucial importance for the future. Important aspects are open systems, native support of independent database management systems, graphical user interfaces, flexible client/server architectures, and last but not least object-oriented development tools;

(f) *vision.* The supplier tries to stay ahead of new philosophies and techniques. Examples of this are the Dynamically Engineered Applications (DEA) and the improvements in implementation and customisation tools.

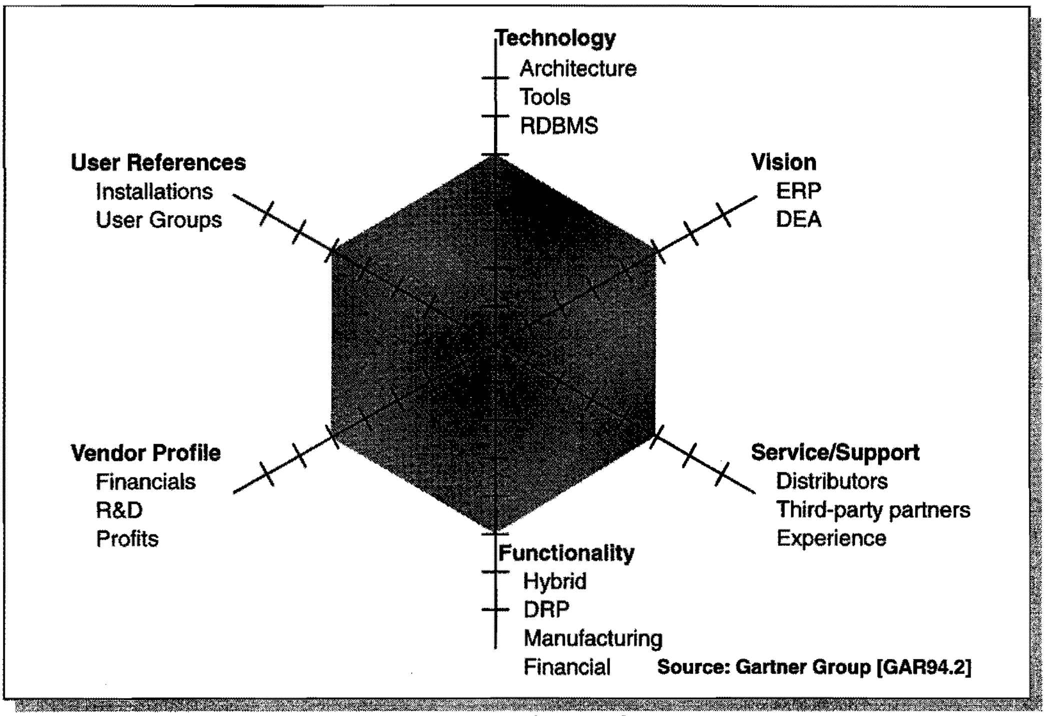


Figure 1.2 - Six-dimensional comparison of Business Information Systems

1116 This classification is mainly focused on international trends and developments as they have been followed in this case by the Gartner Group. The result is a general characterisation of the supply market. In chapter 2, from paragraph 2300 onwards, a pragmatic classification is given that is useful for companies wishing to select a BIS on their own. It concerns a detailed structure of criteria that are of specific importance to a company. The enumeration and assessment of these criteria is part of the selection process.

Considerations on software selection

1117 The previous classification provides a view that can help an enterprise in the selection of a Business Information System. However, the overview is not sufficient for the final choice. The six dimensions were not weighed against the specific requirements, and in particular the functionality has only been assessed in general terms.

1118 Nevertheless it appears justifiable that buying a standard package for manufacturing, distribution and trade businesses is not only a good alternative, but considering the present supply, in-house development of these kinds of enterprise-wide systems is not profitable. This stands in contrast with other branches, like various parts of the services industry (banks, insurance companies), where the supply of software is far more limited.

1119 At present the most important arguments in favour of standard software (instead of in-house development) for replacing an existing information system are:-

(a) the speed with which people can start working with the standard software;
(b) the low costs of purchasing and managing standard software;
(c) the large supply of standard software;
(d) the availability of external support;
(e) the possibility of acquiring new developments in the software and from the supplier;
(f) the enormous size and complexity of in-house software development;
(g) the hypothesis that not every enterprise is different, and that one can learn much from working practices proven by others, without re-inventing the wheel, and by making use of 'best practice' experiences.

1120 This does not mean to say that all these arguments will be applicable in the long term. It is possible that development tools will evolve to the extent that building your own package (on the basis of standard components) will become a viable option again.

1121 Before starting the selection, there should be clarity about the organisation structure and processes that have to be supported, their control and the way in which people wish to work. Important elements are the logistics, commercial and financial economic controls. Fundamental choices and improvement projects with regard to these business principles can have a major influence on the company situation and the extent to which a particular standard package is or is not suitable. On the other hand, one should realise that some types of improvements are technology-driven and will only become possible after the introduction of a modern BIS, especially improvements related to information processing.

1122 Depending on the more or less detailed foundation (the enterprise model), the functional requirements and other criteria have to be deduced. Together they form the specifications and criteria with which the selection process can begin. Concerning the determination of these functional requirements, two streams can be distinguished depending on the situation:-

(a) the specification of an extensive and complete statement of requirements and wishes;
(b) the determination of a minimum number of essential knock-out (selection) criteria.

1123 However, in both cases these requirements have to be placed next to the functionality offered by the nominated BIS. In principle this can be done on paper by means of a questionnaire sent to the supplier. This way the suppliers are formally committed to their answers. An additional test can be a demonstration of the major system functions, usually customised to a company specific case. Here, various suppliers demonstrate their system for a few days.

1124 After an evaluation based on the predetermined criteria, a preference can be shown for the most suitable BIS. Next to functionality, non-functional criteria such as price, support and many others will be taken into account. An attempt has to be made to rationalise this consideration by means of objective criteria.

1125 The selection process is rounded off with the commercial negotiations, which are concluded with a contract. During negotiations functionality plays a subordinate role, this phase is more focused on aspects like price, support (for example, during the implementation), conditions, sizing of the required hardware and maintenance.

1126 Chapter 2 will give a more detailed description of the full selection process, from specifications to the final contract, as well as the important points for concern during the implementation and for the BIS management. While discussing the steps in the selection process, differences in situations will be pointed out, possible variations in approach, various pitfalls, and a 'best practice' approach for package selection.

1127 An important element in the process of making a choice is the evaluation of the BIS functionality based on one's own requirements. This part of the assessment is therefore the subject of this research.

Pitfalls in traditional ways of assessing software

1200 Here, I will expand on the variety of pitfalls and risks involved with the assessment of BIS as part of the selection process. One cannot conclude however that the present methods are bad, or that the choice of a suitable BIS is a question of luck. Naturally, many selection projects are carried out efficiently and thoroughly, with or without the support of an expert (often an external consultant). However, there is a series of less successful experiences or even failures, where only afterwards one began to understand the suppliers product .

1201 The assessment is based on the specification of the functional requirements. Such a specification normally results in a checklist or questionnaire with which the nominated BIS are tested. Problem areas concerning the *statement of requirements* include among other things:-

(a) too great an emphasis on the *present company situation*. Major benefits can be gained by challenging the present ways of working and using the possibilities that information technology has to offer. The search for BIS that cover the current working practices will lead to missed opportunities for improvement and many modifications of the software;

(b) the specification of *unimportant requirements*. Regularly, many requirements are listed that do not directly influence an objective choice, but do worse, they cloud the real issues. This leads to inefficiency in the user organisation as well as that of the suppliers. The collection of requirements are often the fancies of an amateur, features that do not give any benefits (in an economic sense) or are characteristics that can be found in any reasonable BIS;

(c) the *ambiguous formulation* of requirements. In practice many questions put to suppliers can be interpreted in different ways, and with a little creative thinking a supplier can always come up with some kind of 'Yes'. The proper formulation of questions to suppliers is no easy task even for experienced experts.

1202 The reaction to the first two problem areas is to reduce the number of requirements to knock-out criteria, and to leave an important part of the evaluation to a good demonstration of the BIS. The reaction to the third problem area is the opposite: formulate more requirements in a large questionnaire to prevent possible misunderstandings.

1203 On the supply side there is an enormous diversity of BIS from which one can choose, there is something to suit all tastes (see figure 1.1). This present supply, however, also results in a number of problem areas for the selection process:-

(a) due to the large supply of standard BIS, many experts *expect* that a *shake-out* of qualitatively poorer systems will take place. On account of the required continuity in support, it is therefore of great importance to gain insight into this quality. This is not expressed, however, in the traditional questionnaires and neither do demonstrations provide the insight that is required;

(b) because of the *enormous functional breadth and complexity* of the modern generation of BIS, a detailed assessment is becoming increasingly difficult and costs more and more effort. Who can investigate all of these functional areas on alone? This is very problematic in a selection process, when one wishes to compare various alternatives;

(c) in relation to the last point, a *proper demonstration* of all relevant functional capabilities *is hardly possible any more* within an efficient selection process. The key users who have important influence on the right choice run the risk of being swamped with details in those short and overwhelming demonstrations. The users who are used to traditional information systems are more concerned in learning modern user interfaces than with the assessment of the functional capabilities.

1204 The reaction to the first and third point is to bring in (several) experienced experts. However here, too, there is a great difference in quality and experience. One has to trust (external) consultancy to an increasing extent. The reaction to the second point is the extension of the questionnaires to map the very broad and detailed functionality.

1205 Normally, the BIS supplier will react to the requirements through a written response to the questionnaire or by showing the functionality in a demonstration. Naturally, the supplier has a commercial interest and will, of course, try to put his system in the best possible light. *The role of the supplier* gives us the following problem areas:-

(a) a (biased) supplier will *maximise the number of positive answers* ('Yes') to written questionnaires, so that he is not prematurely dropped from the short list. Remember, the supplier will only say what he wants to say, and not always what a potential customer wished to hear;

(b) sometimes during demonstrations *customers are misled* by the supplier, with the exception of reliable suppliers. Some extreme cases are known:-

  (i) fundamentally different versions of a BIS are demonstrated at the same time without this being mentioned raising the suggestion that it is one integrated system;

  (ii) demonstrated screens are switched unnoticed (the 'ALT-TAB' under Windows), while it appears that this is the next logical step in the process flow;

  (iii) almost by definition all the functionality that is not available now, will be available in the next upcoming release;

  (iv) deliberate untruths are told or promises are made where it is known beforehand that they cannot be kept;

  again, this is with the exception of good and reliable suppliers;

(c) the *commercial process* in which the supplier tries, not unreasonably, to build up a relation with the potential customer, fundamentally disrupts the objectivity of the selection process;

(d) the *bias of all parties involved*, sometimes the diversity of informal relations, acquaintances and dependent consultants all result in a disruptive influence in an objective assessment.

1206 A suitable reaction to all these points is almost impossible. Besides being aware of these risks, the support from an experienced but independent expert is about the best precaution one can take.

1207 The quantity of details form a dangerous pitfall, which is partially caused by the size of the functional scope. On the basis of long questionnaires and/or checklists used during demonstrations an effort is made to consider matters objectively. Those involved - users, management, and experts - have a natural tendency to add points for fear of forgetting something. Doing away with questions, however, is extremely hard: „the contributor might take it the wrong way", „perhaps it may become important", or „it can't do any harm to ask".

1208 In an American article from 1992, for example, a full-selection exercise is recommended [SLA92]. Reference is made to „a detailed definition of requirements agreed by the users" and „a detailed comparison of all known software that may satisfy the basic requirements". In practice this means that many thousands of requirements have to be mapped and set against hundreds of BIS available on the market. For this reason such an approach is more an exception than the rule. However, determining several hundreds of requirements and using them for comparisons among five to ten BIS is not unusual. Problem areas with such a '*full-selection exercise*' are as follows:-

(a) the loss of the required *insight* in the BIS because of the large number of details. It is important that insight is gained into the philosophy behind the software, the management philosophy supported by the software, the best working practice supported and the essential functions;

(b) the fading of the essential *distinction* between selection criteria and general questions which are important to understand the system works and what the consequences will be for the organisation at a later stage;

(c) the suggestion of *feigned objectivity* with long lists of questions or check points. Instead, the assessment of long lists of criteria leads to confusion, discussions, and vagueness;

(d) the so-called *weighting factors* suggesting that after determining a factor for each of the hundreds of answers, a total sum simply gives the best choice. However, what is the difference between two systems in a comparison of various BIS where one has a score of 92% and the other has 89%?

(e) the *checking* of the answers and points for each of the individual questions is difficult enough. If you are dealing with lots of questions put to several suppliers then the checking becomes a near impossible task as part of an efficient selection process.

1209 The counter reaction is a more recent trend that advocates package selection on the basis of a limited set of knock-out criteria. In practice this approach nearly always leads to the same choice of a BIS. What does need to be taken into account is the growth of support from the users involved ('quick' but not 'dirty'), when a preference is shown for two or three BIS on the basis of about 25 knock-out criteria. In addition, there is the threat that essential advancement of knowledge is lost among future users through a detailed comparison.

1210 These pitfalls, for instance, become visible in the various *commercially available comparisons of BIS*. Despite their enormous size, the BIS are often limited in specific functional areas. By placing matters under one heading in a comparison, the specific system qualities are lost and blurring of the system identity occurs. The scores are often based on functionality only. The quality of the software and the quality of the data model, the technology used, the user friendliness and the relations with other functionality outside the assessment scope are hardly touched upon.

Potentials of data structures in software assessment

1300 The pitfalls mentioned are best avoided by getting help from experts, who in turn are continuously updating their knowledge in the area of available BIS, including their strong and weak points concerning functionality as well as other aspects. Formal methods and a structured approach to this problem area are lacking. Considering data structures is felt to be a useful extension or alternative by some, but results from the application of data structures or any research regarding their use for the assessment of BIS is not available.

1301 Data structures offer a way to avoid a number of pitfalls in the assessment of BIS, because a good data model is the foundation for system development. Design errors in the data model result in large, unavoidable problems in the further development of a system (i.e., undesirable redundancy or the inability to recognise relationships between entity types). Also, the structural change of the data model further along in the development soon means that much of the programming has to be modified and retested. The fact that the data model is an important factor in the BIS development is underlined by the stringent measures and requirements made by suppliers to protect the secrecy of their data model. Many suppliers are afraid that others, armed with their data model, could make an important step forwards in developing a competitive system. By the way, the attitude of forced secrecy is changing, stimulated by the development of tools that simplify and speed up implementation.

1302 Data structures are designed and determined by means of formal rules and clear schematic techniques. Take for example Bachmann [BAC65], ERM (Entity Relationship Model by Chen) [CHE76], NIAM (Nijssen Information Analysis Method) [NIJS89], and IDEFIX [BRU92]. This makes the specification of the data model more objective and clearer too. In addition, the schematic representations of the data structures are relatively compact (and in any case limited in size) compared to informal detailed textual specifications.

1303 Because data structures are the core of the information system, the data model gives us some idea of the solutions the designers chose in the realisation of a particular function. In the software design process choices have to be made regularly between alternative designs. Such choices are specifically related to the underlying data structure.

1304 Furthermore, another important role of a BIS is to store data and be capable of retrieving data in a large number of ways. In a modern BIS data registration code statement form as much as 75% (even up to 90%) of the functions.

1305 On the one hand data structures can be used for the formulation of requirements, preferably making use of structures and knowledge that is present in standard BIS. On the other hand, one can look at the underlying data structure of the system in question, during the assessment of BIS. In this way the philosophy behind a package becomes visible and one gains insight into (the quality of) the solutions chosen. Therefore, the aim of this investigation is to gain insight into the possibilities and impossibilities of using data structures in the assessment of BIS. A further analysis of this problem area and research objective that follows is given in chapter 3.

Related developments in technology

1400 Despite a long history, the development of Business Information Systems (BIS) will draw to a close in the near future. Besides the growing functionality, both in depth and breadth, the most revolutionary developments are occurring in the area of technology.

1401 Here, a number of important technological developments will be high-lighted with an eye to their influence on BIS. Some of these developments are already in an advanced stage and have become widely accepted by the developers and suppliers of BIS, although various user organisations can be far behind (partially because of the enormous investments in existing technologies). Other technological developments are of a more recent nature and are bringing about a revolution in systems development (and therefore the BIS developers are faced with enormous investments).

Open Systems

1402 Traditionally, the computer market is characterised by a great diversity of systems (system software and hardware), in which developments are strongly driven by the individual suppliers. This traditional diversity has practical implications for users of systems as follows:-
(a) poor exchangeability of hardware components (especially between suppliers);
(b) a limited choice given a certain (vendor) path that has been chosen;
(c) a complicating factor where communication between different systems is concerned;
(d) a poor negotiating position with the supplier;
and all of the above result in higher costs. Not only are the users of system software and hardware confronted with these problems, but the software developers as well.

1403 The solution lies in combining forces and realising standardisation. As a result a number of standardisation organisations arose that work together to a greater or lesser extent with suppliers and/or user organisations. Examples are OSF, ISO, ANSI, IEEE, and X/Open. The result is a spectrum of standards, including both draft and official standards, competing standards, and besides that standards dictated by the supplier market (like MS-Windows). Information systems built from such standard components are referred to as 'Open Systems'.

1404 The objective of Open Systems architecture is important:-
(a) *inter operability* with which the co-operation and communication between information systems is guaranteed by means of standardisation for example, TCP/IP, OSI (ISO), and SNA (IBM);
(b) *portability* with which the operation of information systems is not dependent on underlying hardware and software components and is therefore transferable because of standardisation.

1405 Table 1.1 shows a number of areas in which portability can be realised and includes examples of more or less official standards[2].

1.	User Interfaces	OSF/Motif, X-Windows, MS-Windows
2.	Applicaties	Word, Excel, Lotus123, Wordperfect
3.	Languages	C++, ANSI-COBOL
4.	Tools / DBMS	SQL, Oracle DBMS, Ingress DBMS
5.	Operating Systems	Unix, OSF/1, MS-Windows, Windows-NT, Windows-95, MS-DOS
6.	Hardware	'PC'
7.	Microprocessors	Intel 80x86

Table 1.1 - Areas of portability in Open Systems

2 No statement is made as to whether or not these examples should be considered a real standard, or whether there is a broad acceptance by the market for a product that is completely controlled by one supplier.

1406 By means of complete systems portability, scalability is automatically realised. Thus, we are able to select the underlying hardware and operating systems that can then be sized to an optimally fitting combination. Ideally, an organisation can take its information system from a stand-alone PC to a multi-user minicomputer and then further up to a complex mainframe platform.

1407 The process of standardisation has major consequences for the suppliers market (especially concerning the balance of competition), technological developments and price structures. One good example is the (standard) PC market: developments have taken place so quickly that prices are/were changed nearly every month and the amount of PC software seems almost infinite. Try, for example, to compare the present status of PC hardware and software with that of ten years ago, and then try to extrapolate this for the coming ten years! This development is not only unimaginable, it will have unpredictable effects on the information systems, trade and industry, and society as a whole.

1408 Unfortunately, the use of the description 'Open Systems' does not specify the quality of the underlying standards. This means that besides independent and formal 'de jure' [3]standards, proprietary standards controlled by individual suppliers are also implied. Take for example Unix, to many people often synonymous with Open Systems, which has a large number of supplier specific implementations besides an official standard under the name OSF/1. The co-operation between products from different suppliers is also implied, which is considered to be an independent standard like Windows-NT, MS-Windows, Word by Microsoft, or independent databases such as Oracle and Ingress. *Although not defined purely,* we have chosen to work with the common use, which means that the examples mentioned in table 1.1 are considered to be components of an Open System architecture.

Status of Open Systems

1409 A number of standards within the Open Systems architecture is broadly accepted, as appears from their market share. This standardisation has advanced most in the area of operating systems, where a reasonable degree of hardware independence has been created, see figure 1.3.

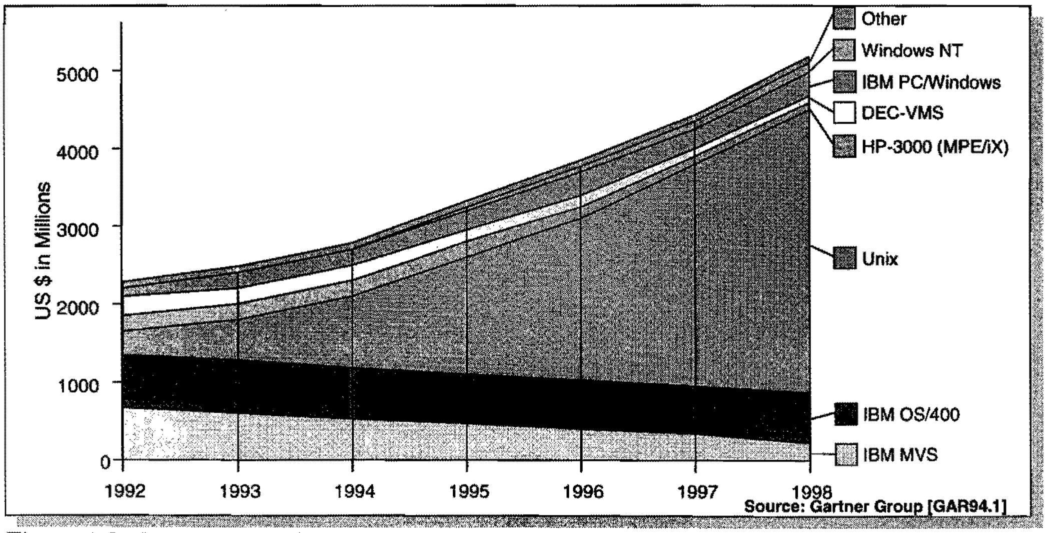


Figure 1.3 - Development in operating systems

3 'De jure' standards are defined by an official (international) standardisation organisation, in contrast with the 'de facto' standards that have no official status.

1410 From this it appears that UNIX is by far the most accepted 'standard' and that Windows-NT by Microsoft, positioned as the major alternative, still has a long way to go. Though things sounded better for Windows-NT towards the end of 1995 (and more especially for Windows'95) than appears from this figure, various independent consultants still have their doubts. In the area of proprietary systems - where hardware and operating systems are coupled one to one - like the HP3000 with MPE/iX and IBM MVS, few new developments are expected. The AS/400, another proprietary system with an enormous customer base will remain stable in absolute size [GAR94/95]. This is particularly thanks to IBM's enormous efforts to transform this platform to an Open Systems architecture. The consequence of operating systems standardisation is that hardware has become a commodity which can be selected independently from the software, purely on the basis of the price/performance ratio.

1411 As a client in a client/server environment (the development in the area of client/server technology is dealt with from paragraph 1419 and onwards), MS-Windows is broadly accepted and its successor Windows'95 will probably gradually become as popular, with heavy competition from OS/2 by IBM. OSF/Motif was considered to be an important competitive (de jure) standard some time ago [GAR94.1], but seems to be losing the battle.

1412 In the area of databases a clear shift towards independent database management systems can also be seen. This technological development will also be considered later in this chapter.

1413 As far as applications like word processors and spread sheets are concerned, it would seem that Microsoft (Word and Excel) is becoming dominant (and therefore sets the standard). Various software developers have formed alliances with Microsoft to integrate their applications with Microsoft's applications.

1414 In the area of communication, TCP/IP Ethernet seems to be a temporarily more or less accepted standard, in contrast with the 'de jure' OSI protocols. Moreover, Novell seems to becoming a sort of 'de facto' standard in the area of (PC-)networks (or will Windows-NT take over in the long term?).

Open Systems in relation to Business Information Systems

1415 The development towards Open Systems has an enormous impact on the development of BIS. The potential market for BIS is becoming much larger and makes high investments justified. Nearly all suppliers of BIS claim to have an Open Systems architecture (whether this is actually substantiated is another matter). While in the early 1980's the number of BIS that worked under UNIX was zero (although UNIX dates from 1968), it was 20% in 1988 and nearly equal to the share of DEC/VAX platforms and the AS/400. In the mean time this share has risen to 65% and is still growing [LOG86/94].

1416 The four most important BIS suppliers on the AS/400 platform (J.D. Edwards, SSA, Marcam, and JBA) are all focused on UNIX. Most of these BIS have reached an advanced stage or are already fully operational under UNIX. Nevertheless, these suppliers are not letting the AS/400 go and are counting on developments by IBM. However, companies that select and implement a new BIS are only advised to use AS/400 if they already have such a platform (and expertise), (see [GAR94.1]).

1417 Another confirmation of this major trend in the direction of Open Systems can be seen in the support provided SAP with regard to the rapidly growing BIS: SAP-R/3. Originally, SAP launched this BIS on proprietary platforms such as MPE/iX (HP3000) and VMS (DEC/VAX). Meanwhile this support has ceased and all SAP efforts are being focused on UNIX and Windows-NT (as a result of AS/400 developments of the this platform may possibly be supported in the long run).

1418 The ability to arbitrarily combine hardware components in an Open Systems architecture based on the price/performance ratio is a consequence of these developments. And though this ideal is often suggested by suppliers, the complexity should not be underestimated. Despite all standards, it is wise for the moment to make one main contractor responsible for the installation and correct running of all components in such an architecture.

Client/server architecture

1419 The client/server (C/S) concept dates from the beginning of the eighties and encompasses far more than just some buzz word, although it is often used that way (which system is not C/S nowadays?). C/S is a philosophy in which tasks are divided flexibly across clients and servers, so that execution is performed by machines that are the most suitable for the job. We are dealing with a distributed software architecture within which clients make requests for the execution of functions, which can be carried out by one or more servers depending on the degree of distribution.

1420 The Gartner Group has defined C/S as „a concept that functionally divides the execution of a unit of work between activities initiated by an end-user program (client) and resource responses (services) to the activity request. Client/server is an application of co-operative processing in which the end-user interaction with the computing environment is through a programmable work station that executes some portion of the application (beyond terminal emulation)".

Typology of client/server

1421 The C/S application model employs a strict division between the tiers of the presentation function, the application function, and the data management function: the three-tier C/S application model. On the basis of this division, the following typology has been developed by the Gartner Group, see figure 1.4, which is considered by many to be normative.

1422 A short clarification of this tier structure:-

(a) *distributed presentation.* Here, we are dealing principally with (graphical) user interfaces on the clients, like X-terminals, which are directly controlled by a central server. Though these clients have some intelligence, such as routines for the drawing of standardised symbols, there is a direct interaction relation between instructions from the server and execution by the client;

(b) *remote presentation.* Here, the logic surrounding the (graphical) user interface can be found on the clients. Examples of such clients are: MS-Windows, X-Windows, OSF/Motif and Windows-NT. This means, for instance, that local software has to be loaded before a user application can run. With respect to the previous variant this leads to an enormous relief for the network;

(c) *distributed function.* Here, the 'application logic' is brought to the client, such as calculation routines, menu control, and complex validations. The logic that applies to the whole organisation, however, remains centralised. One example is the control of the credit limit of customers that has to be guaranteed at all times, independent of which application places orders or receives payments. Traditionally, this 'business logic' is found in the same (by the way most often redundant) software coding as the 'application logic'. To differentiate between both types of logic is one of the most advanced forms of C/S;

(d) *remote data management.* Here, no principle distinction is made between the 'application logic' and the 'business logic', as both can be found on the client. In this shape the server is only used if one requires the database;

(e) *distributed data management* Here, we are dealing with a distributed type of storage of data across several servers. By means of an extremely advanced (but complex) type of distribution, an effort is made to improve the communication, speed, availability, and management. Within this form, a distinction can be made between homogeneous and heterogeneous architectures, in which in the latter case, the various database servers operate under different database management systems.

Within this complex type of the C/S concept, a large number of other characteristics play a role, which are beyond the scope of this investigation.

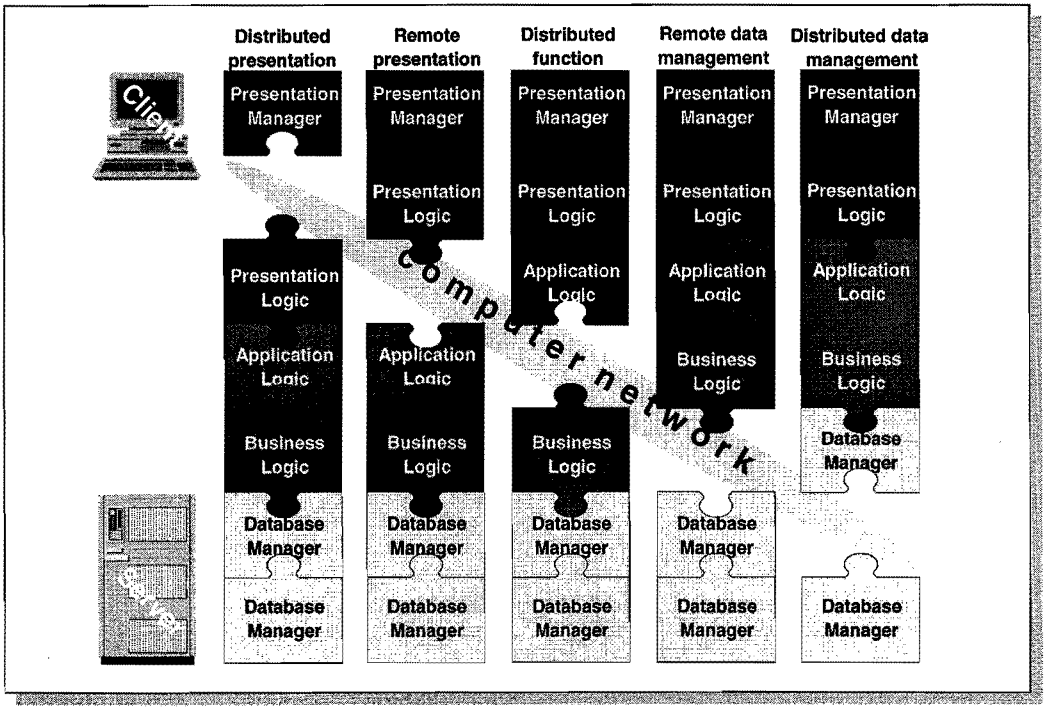


Figure 1.4 - Client/server typology

1423 At the heart of these types of C/S we have the computer network, which lies on the dividing line between the clients and the servers and takes care of the communication between the components. Figure 1.4 suggests that there is a dichotomy between the role of the client and server: the so-called two-tier concept. However, distribution combinations of presentation, application and database logic are possible too. Such combinations that lead to a trichotomy for example, distributed function and distributed data management, lead to a three-tier concept.

1424 Other kinds of components that can occur in an extensive C/S architecture include:-
(a) printer servers that are specialised in the control of a collection of printers that can be used by the entire network;
(b) batch servers that can carry out sizeable and/or regular activities without burdening other network components;
(c) communication servers that for instance control the communication between different sites.

Characteristics of a client/server architecture

1425 One of the characteristics of a C/S architecture is to relieve load on the central mainframe by means of downsizing. Through a combination of various specialised components - like presentation servers, database servers, a modular architecture arises. For each of the components an optimal choice can be made so for example, and an older 80286 PC could still be used as a printer server.

1426 Such a modular structure also leads to flexibility and scalability. Thus, one component can be replaced by a larger system, tasks can be split across several systems or tasks from different systems can be combined on one system. The result should lead to considerable reduction of costs.

1427 The present generation of C/S architectures, however, also has a number of problems or risk areas to contend with:-
(a) the degree of standardisation to make different components work together, particularly if they originate from different suppliers;
(b) the availability of real C/S applications. Many suppliers claim to supply applications that completely support C/S, but in practice many do not go further than remote presentation (often under MS-Windows);
(c) the stability and vulnerability of the computer network and the whole complex of components.

1428 A C/S architecture can become extremely complex and as a result the burden of system management can increase enormously. This is an important reason why in many cases the cost reduction of downsizing lags behind expectations. The integral costs may even increase as a result of this complexity. For further details see the large number of articles published on this subject.

State of the art concerning Business Information Systems

1429 In the largest part of the present BIS, the C/S concept does not stretch any further than remote presentation or remote data management. Only a limited number of BIS, about 10% according to [GAR94/95], support some form of distributed data management. However, this group is growing fast, partially driven by the development of database management systems. The most advanced forms of C/S, distributed function, are hardly supported. As was indicated earlier, this is mainly caused by the lack of suitable development tools, thus many of the larger developers of BIS (such as Baan, JBA, SAP, and SSA) are forced to build their own development tools.

1430 According to the commercial presentations of some BIS, the advanced forms of the C/S concept are often supported. In reality we are often only dealing with a part of the total BIS, being the part that has already been rewritten in a new (object-oriented) environment. Figure 1.5 shows the essentially different C/S architecture of two BIS.

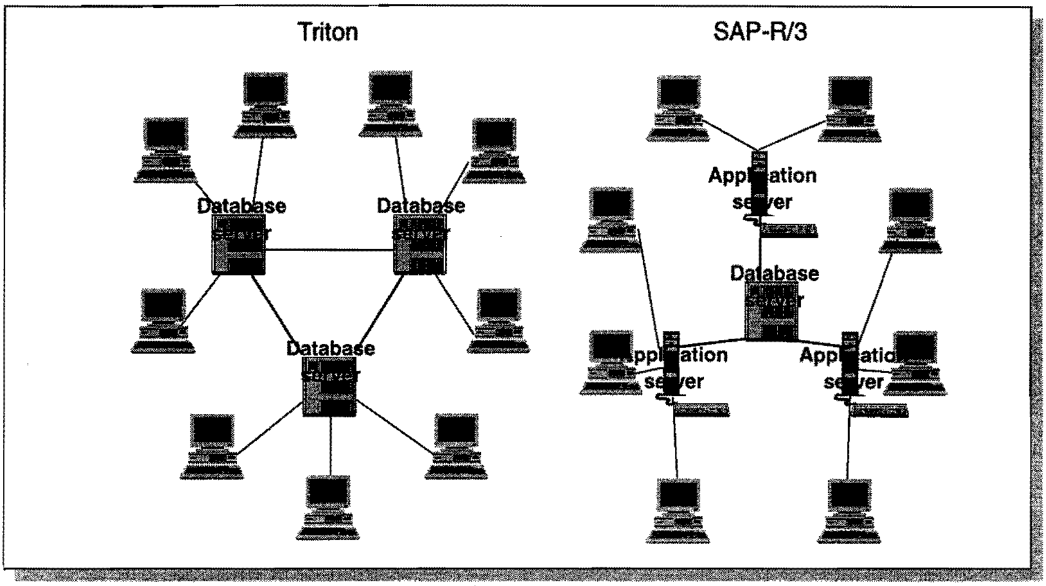


Figure 1.5 - C/S architectures of Triton (release 3.x) and SAP-R/3 (release 2.x) compared

1431 As can be seen in this figure, the most extensive form of Triton has separate databases per site, which are run on different database servers. In the layer above, the different database management systems look after the distribution of data over various servers for example, the replication of mutual customer data or item data. Without doubt such a structure places quite a burden on the network. This is a C/S architecture of the type two-tier 'distributed data management'.

1432 SAP-R/3 features a totally different C/S architecture in version 2.x. An explicit distinction is made between presentation, application, and database servers. Besides the fact that these server functions can run on separate hardware, these can also be combined on to one machine. This is a C/S architecture of the type three-tier 'remote presentation' combined with 'remote data management'. In the mean time this architecture has been extended with another two types in version 3.0 of SAP-R/3, i.e. the 'distributed data management' in which tables can be (horizontally and vertically) segmented across multiple database servers. The second type is the multi-site data structure which was also described for Triton. Here, the link is realised by means of the ALE concept (Application Link Enabling) developed by SAP.

1433 One important consequence of the continuing development of C/S architectures, as well as the development of ERP-software, is that networks have become the essential element. Probably computer networks will even become the most important bottle-necks in these developments.

Graphical User interfaces

1434 Graphical User Interfaces (GUI's) date back to the beginning of the eighties, and were first introduced on the market by Apple on machines like the 'Lisa' and the 'Macintosh', which still is a modern machine today! The original idea and the development of the first prototypes come from the laboratories of Xerox, who sold them to Apple, doubting their usefulness. For a long time the Apple Macintosh was a unique product with its own niche market. Parallel to the rapidly growth of the PC and DOS market, there has been a small group of fanatic fans of this optimal integration of application software, operating system, and its uniform user interface.

The importance of GUI's

1435 Until recently these user friendly interfaces were dismissed as a fad. One important argument was that during mass data entry such an interactive tool would be counter-productive. Meanwhile a change has set in, probably due to habit on the one hand and on the other due to the introduction of principles like business process re-engineering. The latter has certainly led to the reorganisation of many business processes, in which work no longer piles up for batch processing, but has resulted in the processing of data directly within each business process. From research it also appears that productivity increases with GUI's. The advantages recorded are faster work, less mistakes, and reduced fatigue.

1436 Now a GUI as a front end has become a selection criterion and software packages without GUI support hardly manage to get a foot in the door.

The development of GUI products

1437 Besides the trusted Macintosh (which should not be discounted for the future), this success was copied by Microsoft with MS-Windows. After a difficult start, partially caused by the hardware requirements, version 3.x brought an actual breakthrough. The labelled successor and as far as Microsoft is concerned the 'de facto' standard is supposed to be Windows'95. Next to Microsoft's success, IBM has been fighting for their alternative: OS/2. With their present 'Warp' version they are trying to win a market share from Microsoft. Considering IBM's enormous marketing efforts, OS/2 should in future not be ignored.

1438 Furthermore, we have OSF/Motif, a GUI developed by an independent standards organisation. OSF/Motif is coupled to X-Windows which itself only looks after window manipulation and is not a graphic interface. OSF/Motif was also believed to be an important GUI product for the future according to the Gartner Group.

1439 Finally, we have the AS/400 which lacked a GUI until recently. A product called GUI/400 was developed by the Dutch company Seagull can provide a graphical interface on top of existing software. The product has been adopted by IBM and is distributed under a world-wide licence. However, considering the necessary developments that the AS/400 platform is undergoing (towards an Open Systems architecture), this can only be seen as a temporary solution.

Consequences for the Business Information System

1440 Nearly every supplier of standard Business Information Systems is coupling (in so far as it has not already been completed) their software package to one or several of the GUI's mentioned above. One of the first software packages that was completely based on a GUI was SAP-R/3. This is probably why this package became so successful so quickly. In contrast to the existing software packages, in the development SAP chose not to support character terminals at all. The moment the new package became available (1991) it precisely fitted the growing interest in the market for GUI's.

1441 For existing software packages, the support of both kinds of interfaces is still required to give users the time to accommodate. However, in the long run (a few years) it can be expected that the support of character terminals will disappear completely. The differences in architecture are so large that software packages cannot possibly make optimal use of both types of interfacing, unless compromises are made in the design or two completely separate versions are developed (and supported).

1442 Besides the technical standardisation in the area of user interfaces, hardly any steps have been taken towards *standardisation at a functional level.* In the area of screen lay-outs and functional interaction between the user and the system (the GUI) there also is a growing need for unambiguous symbols. Especially for users confronted with software packages from different suppliers. It is possible that this type of standardisation will be boosted by the developments surrounding object-oriented systems.

Computer Aided Software Engineering

1443 Computer Aided Software Engineering (CASE) covers the spectrum from information planning through the specification, the design and the development of information systems, including maintenance. It encompasses a collection of tools that support that entire life-cycle of information systems. Besides CASE, similar terms like IPSE (Integrated Project Support Environment) and AWB (Analyst Workbench) are used in development environments. In practice no real distinction is made between these tools, so that we have chosen to use CASE in this context. Essentially CASE tools support the highest abstraction level in software development. Most tools are highly graphically-oriented and models that have been developed can be altered quickly and easily. In principle it is possible to develop and maintain efficient and effective information systems using the integrated CASE tools [VON88].

Positioning of CASE tools

1444 An integrated CASE environment is usually based on the support of one or more development methods, like SSADM, Method/1, or IE (see figure 1.6 for the latter). The various tools represent the techniques within such a development method.

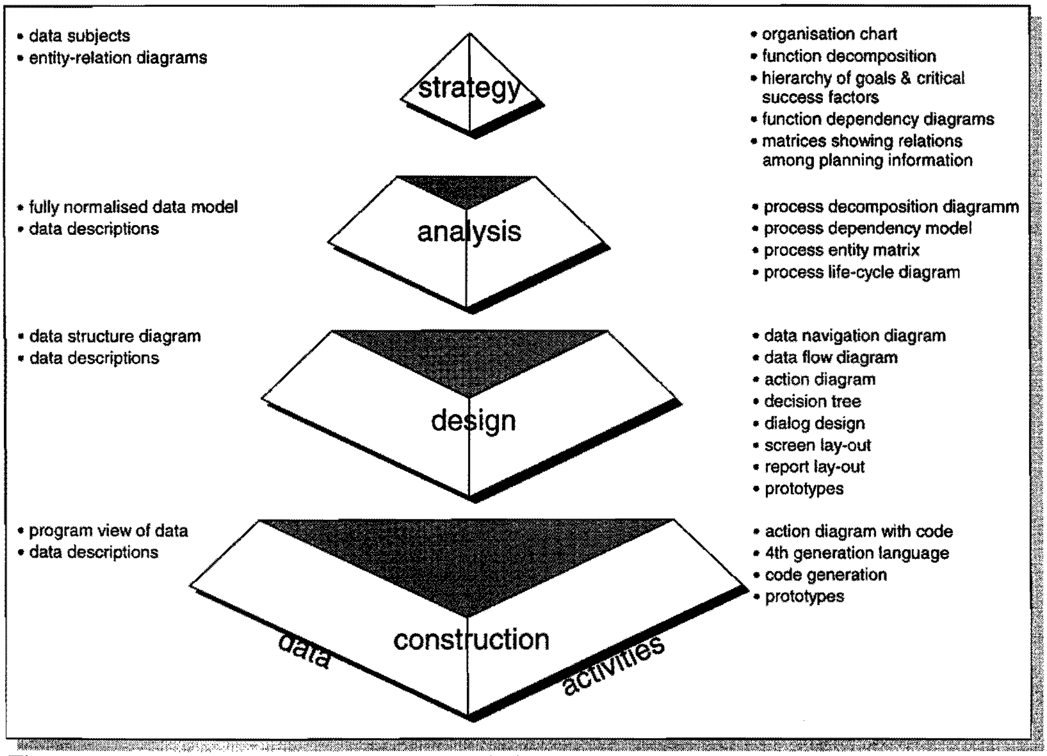


Figure 1.6 - IE method and incorporated techniques [MAR86]

1445 Among other things, CASE tools are used for the following [VAR93]:-
(a) the drawing of (design) diagrams such as Data Flow Diagrams and ERM schemes;
(b) the monitoring of the consistency both within a scheme as well as between the various diagrams and between models of the various tools;
(c) the management of all data, definitions, and models in a so-called repository;
(d) the documentation of resulting information systems;
(e) the generation of executable software code.

1446 The main characteristic of CASE tools the integration or linkage with a 'repository'. This repository, also referred to as a data dictionary, is a central database in which, among other things, various models developed using these tools are stored. In the absence of a repository, one could presume that we would only be dealing with basic development tools like higher programming languages.

1447 With this repository as a kernel, the various CASE tools can be divided as follows [WIJE93]:-
(a) *vertical tools* restricted to a certain phase in the life-cycle of system development, which can be divided into:-
  (i) upper CASE tools aimed at the information planning and design stages;
  (ii) lower CASE tools aimed at implementation, including code generators;
(b) *horizontal of cross life-cycle tools* for instance used for project management ;
(c) *service tools* for the management of the repository, the distribution and integration across several developers, and version management.

1448 In the figure below this division is represented schematically. In principle integrated CASE tools cover the entire life-cycle completely.

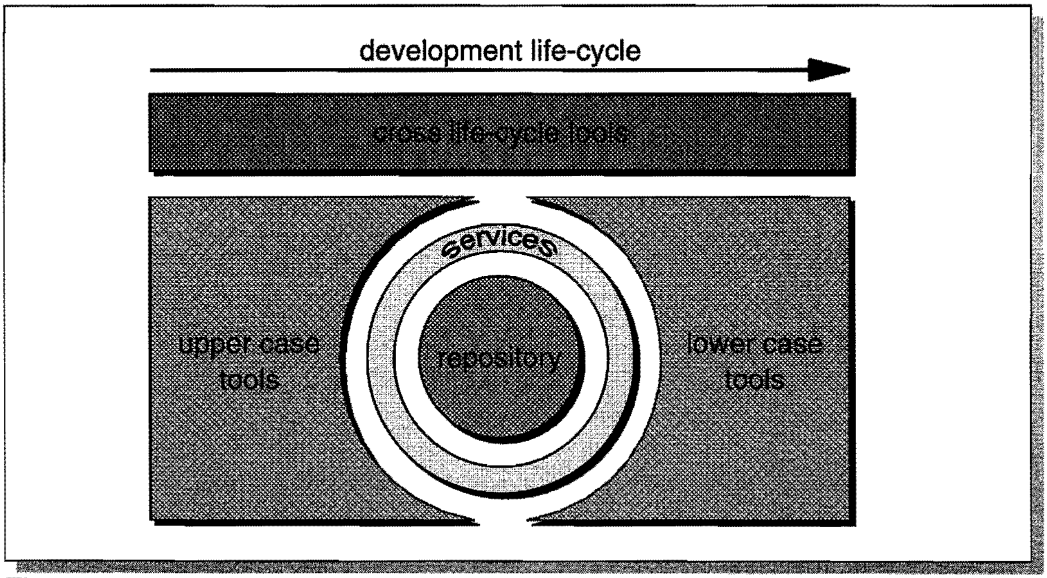


Figure 1.7 - Structure of Integrated CASE tools

1449 Examples of general 'open' CASE tools (see paragraph 1402 and further on 'Open Systems') include IEF (Information Engineering Facility) by Texas Instruments, ADW (Application Development Workbench) by KnowledgeWare, and SDW (System Development Workbench) by CapVolmac. These are used mainly for developing in-house information systems.

1450 CASE tools are of course intended (and highly suitable) for the development (and maintenance) of information systems. For standard BIS this phase has been completed however and maintenance too is more a problem for the supplier. However, if BIS are developed with the aid of CASE tools this will certainly be reflected in the quality, the documentation and the adaptability. Some important BIS that were developed using CASE tools are:-

(a) CIIM by Avalon, developed with the Oracle Case;
(b) BPCS by SSA, developed within their own CASE tools, i.e. AS/SET. AS/SET is now widely available;
(c) J.D. Edwards by J.D. Edwards, developed with their own WorldCASE;
(d) Triton by Baan, developed with the help of their own Triton Tools.

1451 The fact that the suppliers of BIS develop their own CASE tools does say something about the availability. Presumably, the weak spot of many lower CASE tools and services lies in the dependence on a hardware platform and database for the generation of program code. If we are interested in documentation aims, the supply of upper CASE tools is sizeable and nearly all standard BIS make use of them to a greater or lesser extent.

1452 A particularly interesting aspect in the acquisition of a standard BIS is the use of CASE tools with which one is easily capable of customisation[4] the result. Instead of arranging the BIS by means of parameters, where one often has to judge and fill in maybe thousands of parameters, the CASE tools offer far more degrees of freedom and generally provide far more insight. Areas where customisation with CASE tools can be easily applied are:-

(a) screen lay-outs;
(b) navigation through the BIS;
(c) data structure: adding new attribute types, entity types, relationship types and stored procedures (see paragraph 1460 and further on 'Database Management Systems');
(d) control and security.

1453 With this customisation we should be careful that we remain fully capable of moving towards new releases of the standard BIS. The supplier developments never stop and besides that possible errors have been removed in a new version, new releases often offer interesting functions and technological improvements. Take for example the user friendliness and client/server architecture.

1454 An entirely different area of application for CASE tools in relation to standard BIS is the specification of the (logistical) control structure and the functional requirement as a basis for the selection. Next to general scheme techniques, the support of techniques like GRAI and IDEF is interesting in the area of manufacturing management [GAV94] (these will be looked into in more detail in chapter 2). However as part of integrated CASE tools, the support of these techniques is rare.

---

4 Customisation is a collective term for the adaptation of a standard BIS for situation specific requirements without changing the source code, but by means of tools that are specially intended for this purpose.

1455 In the above we briefly discussed the customisation of a standard BIS with the aid of CASE tools. A much more sweeping approach is to work on your own (further) developments on the basis of a so-called CASE template[5]. Such a template is in fact a working BIS that is completely based on CASE tools. One important difference is that whereas a normal BIS will try to cover as broad a market segment as possible, and therefore will offer quite a lot of unnecessary functionality, these templates offer a minimum of functionality as the basis for an optimal fitting system. Actually, by using a template, one gets off to a flying start with the (further) completion of one's own BIS with a perfect fit.

1456 The power of this approach therefore lies in the quality of the CASE template and of course the quality of the CASE tools themselves. Essential to the template approach is the notion that one starts of with a minimal option that is to be extended, while most BIS are based on a maximum option that is to be trimmed down by means of parameters [HOF93].

1457 Due to the limited supply, in most situations this approach does yet not offer a real alternative. Most BIS are not developed on the basis of integrated CASE tools and moreover they are functionally too complex. The supply of real templates is limited, possibly because of the limitations of the CASE tools. Besides special requirements like code generation, the cost factor starts to play a role in such extensive tools.

1458 One example of a template is Implementor [BUT92], in which the developer himself has developed a limited number of CASE tools. It is difficult to decide whether to continue development on the basis of CASE template or to opt for a (completely) standard BIS. Arguments against the development on the basis of a template (on one's own) are the build up of expertise, the investment in the (often expensive) CASE tools, and that one can no longer upgrade towards newer versions. On the other hand, one does have a BIS that is a perfect fit.

1459 By the way, there lies a danger in the complete adaptation of the template to one's own situation, being that one may accidentally choose relative inefficient or ineffective working practice. By adopting the restrictions of a standard BIS, one may be forced to adapt the current working practice to a kind of proven 'best practice'. This danger is demonstrated in [HOF93], where a description is given of major adaptations to the ledger administration (while this is just about the most crystallised functionality in a standard BIS). Nevertheless, this option was considered to be a great advantage!

Database Management Systems

1460 The development of database management systems (DBMS) started in the sixties. For a long time these DBMS were based on a physical organisation of the data according to a hierarchical or network structure. Until the late eighties, these DBMS were able to remain commercially competitive. A number of typical examples are IMS by IBM, an hierarchical DBMS, Total by Cincom, Image by Hewlett Packard and IDMS by Cullinet, all being network DBMS. IDMS is also an example of an implementation of the official Codasyl standard for network models [DBTG71]. Though all these DBMS are outdated, IMS is still the foundation for many operational information systems.

---

5 The term template is also used for a stripped version of a BIS, such as Triton light, where there is a series of pre-programmed parameters settings that enhance the implementation. Besides such stripped versions, parameter settings or models for a specific industry sector (like automotive) are offered as templates.

1461 In 1970 the relational model was developed theoretically and first described by Ted Codd [COD70]. This model is completely based on algebraic operations and on the calculus-based languages. SQL is the most standardised example of this. Though the first DBMS that supported this model were already developed in the mid-seventies (by Ingress and Oracle among others), it took more than ten years for them to be applied on a larger scale. The speed of the available hardware was a bottle-neck for a considerable time. After that, the relational model was further extended among other things, with semantic elements, see the Entity Relationship model by Chen [CHE76] and the Semantic data model by Hammer and Mcleod [HAM80]. The relational (and semantic) model are now by far the most applied model in DBMS.

1462 Next to the DBMS mentioned, various file management systems were (and still are) applied (such as Btrieve under Novell or Bbx-basic files under UNIX), where the application software is responsible for guarding the structure and consistency. Only now is this simple but cheap technique slowly disappearing.

Developments in database management systems

1463 The use of relational DBMS seems to be common practice nowadays. There are further developments, however, within the relational framework. Typical extensions include a diversity of new types of attributes, like document text of variable length, image and sound. Besides that, more and more functionality is being defined in the data structure, under the control of DBMS. Examples of this are the 'stored procedures', which are executed once a certain condition arises in the data structure (the 'trigger').

1464 However, we can only speak of the complete integration of data and functionality, in the so-called object-oriented DBMS. Though some of these DBMS are now commercially available, they are still primitive. Later in this chapter we will discuss object orientation.

1465 Other important developments are taking place in the area of tools surrounding the DBMS. Both the DBMS supplier and third parties are (further) developing tools for query and reporting (for example, Corvu, a graphic query / reporting tool for Sybase). In addition, developments towards fourth generation languages and CASE tools are also taking place (for example, SQL-Forms by Oracle and Momentum, an object-oriented development tool by Sybase).

1466 Finally, tentative practical testing of distributed databases has started, which until now had mainly existed as a theoretical concept. The reasons for moving towards distributed databases lie in the following:-

(a) the decentralised nature of their use. Much data and therefore its usage is centred around company divisions or sites;
(b) the increased availability and reliability. Through the introduction of redundancy (replication) of data, by placing data on several servers, the same information can be found in several different places. Naturally, the DBMS has to guarantee the integrity and consistency of the redundant data. Moreover, the spread of data across several servers will make one less dependent on the availability of one specific server;
(c) the improved performance. The data is located in the place where it is used the most.

1467 The technology to make optimal use of distributed databases is characterised by the following [ELM89]:-

(a) *transparency*, so that the application (and the user) does not have to worry about where the data is located;

(b) *replication* under the control of the DBMS, so that availability and speed increases. Here, complicating factors have to be taken into account like synchronous updates, consistency control, and recovery;

(c) *horizontal and vertical segmentation of tables*, with which one table can be divided across several database servers;

(d) *heterogeneity*, so that each database server works under its own DBMS without this influencing the logical data structure across the database servers.

1468 From this it should be clear that the speed and reliability of the technical infrastructure is of crucial importance when working with distributed databases.

Application in standard Business Information Systems

1469 Various suppliers of BIS have built the tools (like the DBMS) with which the BIS was originally developed. One typical example is Baan Company. Now most suppliers have turned to commercially independent DBMS, sometimes next to their own DBMS. The DBMS applied by far the most in standard BIS is Oracle. A clear second is lacking, as it is more or less shared by Sybase, Informix, Ingress, and Progress, next to the xBase variants for the smaller PC platforms. Some distance behind them we find DB2 by IBM (as an applied DBMS in standard BIS), and a possible newcomer is probably the SQL*Server by Microsoft. Besides these independent DBMS, the internal AS/400 DBMS (DB/400) should not be forgotten.

1470 With regard to the support of DBMS in standard BIS, we can discern two more recent trends (see also [GAR94/95]):-

(a) the shift from a single DBMS support to multiple DBMS. Nearly all suppliers are striving to support Oracle, Sybase, Informix and Ingress. Even the developers of these DBMS are busy trying to support the competing systems within their own development environment. This will make the previously cited heterogeneous databases possible;

(b) the distinction between native and non-native DBMS use. In the non-native approach to DBMS, a translation is made during the execution of function calls from an application to the specific syntax that the DBMS requires by means of the so-called middle-ware. By using a limited number of DBMS functions in the application, it is relatively easy to support several DBMS. In the native approach this middle-ware is absent and DBMS functions are directly called for from the application. The disadvantages of the non-native use (and therefore the advantage of the native approach) are:-

    (i) the delay factor caused by the DBMS drivers (the middle-ware), because a translation has to be made every time;

    (ii) the impossibility of optimal use of the specific features on each of the DBMS used. In the non-native use of several DBMS, only database functions can be used which can also be translated for all other DBMS. The rest has to be solved in the application;

    (iii) a larger part of the functionality than necessary lies in the application, so it is dangerous to access the database directly instead of via the BIS. If eg, locking or consistency checking is controlled by the BIS instead of by the DBMS, then the integrity of the database can be endangered if the same database is modified by external tools (this problem does occur with a number of BIS in practice!).

1471 BPCS for instance has found an elegant solution for the native support of several DBMS. All specifications of the BIS are defined in a repository (see paragraph 1443 and further on 'Computer Aided Software Engineering') and are independent of the applied DBMS. A DBMS-specific generator generates all final code. For each of the DBMS to be supported, a proper generator that allows the maximum use of the options of the DBMS has to be built leaving the rest to the application.

1472 The development of the distributed databases has already been mentioned. Currently, little experience of distributed databases has been gained on operational BIS. The most important suppliers, however, are very active with this development, particularly in relation to multi-site structures (separate database servers per site). Triton now offers an infrastructure in which the distribution of data is controlled outside the DBMS used. SAP supports the horizontal and vertical segmentation of tables on the basis of Oracle in version 3.0. Driven by the necessity to support complex multi-site organisation structures, this development will continue in the coming years.

Object-oriented technology

1473 The most important technological development has to be object-orientation (OO). The objects being central to this approach are:-
(a) physical objects eg, materials, machines or workers;
(b) information objects eg, a bills of material or routings or orders;
(c) control / decision objects for example, objects for the evaluation of a state, the execution of an algorithm, or action taking.

1474 The object technology is in fact a synthesis of a data-oriented and a function-oriented approach. Objects are defined in classes (an object type). Five object classes are represented in figure 1.8. Objects (specific instances, representatives of the class) exchange data through so-called messages, three of which are drawn in this figure. These messages activate predetermined methods (kinds of functions or procedures defined within the framework of an object class) that define the possible interaction of an object with the outside world. The definition of object classes can occur hierarchically, in which subclasses inherit the features of the super classes. An example is given in the diagram. This finalises the summary of terms. For more information the reader is referred to [SPU94], [TAY93], and [TAY95].

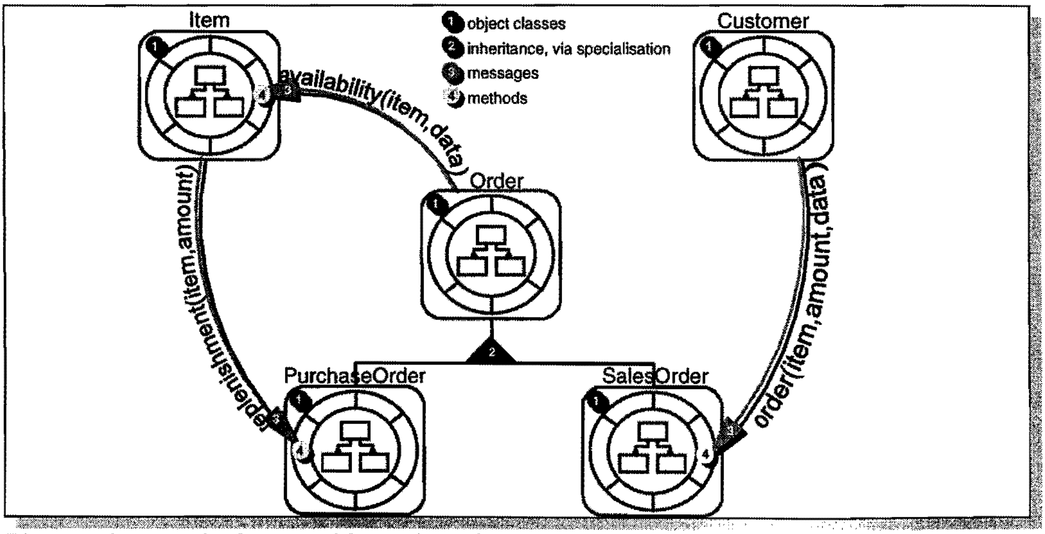


Figure 1.8 - Terminology in object orientation

Advantages of object orientation

1475 The advantages that can be achieved with object orientation seem to be great, both for the developer and the software supplier, and maybe even for the potential users. For instance, the independent structure of object classes, for instance, matches well with a real client/server architecture, where it would be possible to freely determine by which (business object) server specific objects are served [SPU94].

1476 A brief overview of the most important advantages:-

(a) *faster development of information systems.* This can be especially achieved through:-
   (i) *reuse of existing object classes*, which does of course mean that they have to be thought through, carefully designed and developed;
   (ii) *assembly of standard object classes* (as components) both self-developed but also from third parties (valuable libraries of object classes are added by suppliers of OO-tools);

(b) *higher quality* of the systems developed which can be achieved through:-
   (i) *reuse of proven object classes*;
   (ii) *encapsulation and hidden information* so that the internal behaviour of an object does not influence its environment and vice versa;

(c) *simplified maintenance* which can be achieved through:-
   (i) *inheritance* of characteristics from super classes, which therefore only have to be defined and maintained once;
   (ii) *encapsulation and hidden information*, so that in modifying or replacing an object class, the only concern is the unambiguous communication protocol (via standards) with the outside world;
   (iii) *forced modularity* that is inherent of the thinking and working in object classes;

(d) *selective subcontracting* of the development of object classes which leads to:-
   (i) *specialisation of object classes.* Everybody develops object classes that they are good at;
   (ii) *optimal buying/making decisions*, namely for each object class at a time.

1477 These advantages lead to lower costs in the long run, if they are properly taken into account in the initial development. The crucial factor in this is the standardisation of object classes, their interfaces, and an open architecture within which this diversity of object classes can communicate, see paragraph 1480 and further.

Developments in object-oriented technology

1478 The OO-approach dates back to the sixties and was used (and still is) for the development of simulation models. The first programming languages based on this are Algol'68 and Simula. Until the late eighties, OO-technology was restricted to these simulation languages. Only later, particularly driven by the (object-oriented) graphical user interfaces and the integration of multimedia applications, OO-technology really took off. Actually, fully fledged OO-development tools, in which complex BIS can be built, have only recently become available. This does not hold yet for the OO-DBMS, the present generation does not seem mature enough and people are generally advised to make use of the relational DBMS[6].

---

6 Considering the enormously rapid developments - especially in the area of object orientation - such an assertion may already have become outdated by the time this is read.

1479 And though OO-technology has only recently attained maturity (or perhaps not quite yet?), it is viewed by experts as the most important technological revolution. In an investigation of Object Technology from 1993 [OBJ93], the year 1996 was predicted to be turning point in the revolution (which indeed seems to have become reality). In the year 2000, 80% of all businesses (in the USA) will have transferred to OO-technology. This estimate may seem optimistic, but various developers of standard software are already redeveloping their software.

1480 As stated earlier, the availability of standards is an important element in the success of OO-technology. This was already recognised in 1989 by a number of founders of the Object Management Group (OMG). This organisation which is successfully involved with the development of standards in the area of OO-technology, is now being supported by more than 400 organisations from information systems suppliers, software developers, and consultancy organisations to user organisations. The objectives of OMG entail the definition of industrial guidelines and object management specifications that have to provide a common framework for the development of applications. Conformation to these specifications makes it possible to develop a heterogeneous application environment, in which all important hardware platforms and operating systems can be supported [SPU94].

1481 Whether this ideal will ever be achieved still remains to be seen. The development of standards is one thing, but obtaining a broad base of support and commitment is quite another (compare this to how long it took for UNIX to become a so-called standard, see paragraph 1402 and further on 'Open Systems'). The conformation of suppliers to such standards is in opposition to their natural role to maximise the dependence of users to their product.

Developments in standard Business Information Systems

1482 In the mean time most of the larger developers of BIS have started to rebuild their systems, making use of OO-technology. In practice this indeed means a complete redesign leading to functionally completely new products. See for example the new financial modules of BPCS: Configurable Enterprise Financials (CEF) by SSA. These modules are not even remotely like the old software written in RPG. Through their extensive functional possibilities, they even hold a market position as a financial system, separate from logistics modules, which have to compete with specialist systems.

1483 Some other more advanced OO-developments that can be found with suppliers of BIS include:-
(a) Protean by Marcam, the successor of the quite famous Prism;
(b) Adage from the supplier of the same name, an entirely new development;
(c) PIUSS Penta by PSI, the successor to PIUSS-O;
(d) System'21 by JBA, the successor to Business 400.

1484 Beside these suppliers, various developers are busy with rebuilding, which will take a number of years to complete. For many this means that a choice was made for a gradual transition, and that the OO-modules will be released onto the market in a number of phases. Considering the enormous investments, a number of suppliers will not be able to carry the burden of this development, while others will simply be too late. The expectation is that on the one hand there will be a shake-out of BIS suppliers (see also [GAR94/95]), and that on the other a number of completely new players will appear on the market, as a consequence of the rapid development of new OO-tools, who are not hindered by the transition from an existing BIS or group of customers.

1485 The fact that the available OO-tools (and database management systems) have not finished their evolution is an important influence on this development. At present, little is clear about the uniformity and standardisation of tools, so that choices made now may stand in the way of developments in the future (if it turns out to be a bad choice in retrospect). Thus, various developers have been forced to build their own OO-tools (for example, SSA and JBA) and will have to continue to develop them. This means extremely high investments which are not immediately for the benefit of the BIS functionality. The quality and openness of the tools used will in the long run be essential for continuity.

1486 The real advantages for the user organisations, however, are not directly visible, with the exception of the additional functionality as a result of the redesign. The first advantages (see the enumeration in paragraph 1476) are mainly for the supplier. Possibly, these advantages may be turned into lower purchasing costs in the future, but the high investments will first have to be recovered. The advantages that are interesting for the user organisations will only arise when the standard open architecture becomes available, where specialised object classes from different suppliers can be placed within the framework and can communicate via (semantically) standardised messages.

1487 If such a standard architecture becomes available with a sufficient supply of independent business object classes, the monopoly position of the larger BIS suppliers will disappear and a revolution will take place on the supplier's market. It is already clear that the number of BIS based on OO-technology - driven by the technological developments - will steadily grow.

2 | State of the art in software selection

2001 The selection of standard Business Information Systems (BIS) from the large supply available on the market, is an activity that many consultancy firms and consultants are involved in. The combination of all these experiences gives a 'best practice' approach for selecting a BIS. This chapter describes the background, an approach, and the context of the selection of BIS. First and foremost it is a description and amplification of observations from these practices and not the result of fundamental research.

2002 From paragraph 2100 onwards I shall go into the varying organisational situations, and the resulting objectives and points of departure. Often these are contained in an information plan.

2003 Next, the selection process itself is expanded from paragraph 2200 onwards. This starts off with the determination of the business concept. After that, a method is described with a number of optional elements. I will also go into the critical success factors in selection and the various areas of expertise that have to be covered in this process.

2004 Then beginning at paragraph 2300, the different kinds of requirements that can play a role in the assessment of BIS are discussed. After a closer look at the functional requirements, a number of warnings are given for the assessment of these types of requirements.

2005 Paragraph 2400 and onwards deals with the large influence the choice of BIS has on the implementation process. This can be retraced to the way in which the degrees of freedom of the standard BIS are customised and the way in which a BIS is adapted to the organisational requirements. Here, use is made of a typology of customisation types.

2006 Finally, paragraph 2500 and further goes into the developments within the supplier market, the functionality, the technology, and ways of implementing BIS. These developments are of great importance in creating the right pattern of expectation when choosing a BIS.

Differences in situations and objectives

2100 Before starting with the selection and implementation of a BIS, one should realise that many situations, objectives, and starting points can be distinguished, if only as a result of the size and type of enterprise. This typical situation is normally explicitly reflected in the information management strategy. Even if this strategy is not explicitly formulated, in practice certain (implicit) choices are made.

2101 In order to give some idea of possible situational differences, I have worked out a few different perspectives:-

(a) strategic objectives of the enterprise;
(b) size of the enterprise;
(c) supply of systems for a particular business type.

2102 Finally, some examples are given of internal conditions or choices that, as a part of the information management strategy, influence the selection process.

Information technology as a competitive advantage

2103 The role of information technology (IT) is often regarded as a necessary tool for the support of an organisation and its business processes. From this perspective, IT is viewed as a cost burden and a search is made for the cheapest possible system (on all integral costs being taken into account). This results in the traditional cycle of purchase (or development) and implementation after which the most static situation possible is maintained for the maximum number of years. After such a cycle, which sometimes lasts more than ten years, the cycle starts again with the purchase or development of an entirely new information system.

2104 Another approach is aimed at the strategic use of information technology with which the company can:-

(a) distinguish itself from the competition;
(b) hold onto its customers and influence its negotiating position;
(c) maintain its position with respect to other suppliers;
(d) keep the competition at a distance;
(e) offer new or improved products.

2105 Examples of this are: interfacing or integrating with the information systems of customers and suppliers, relocating activities in the logistics chain (for example consignment stocks and forecast reporting) and fundamental product or process innovations, as a result of among other things successful business process re-engineering projects [WEE93].

2106 Such a strategic choice has consequences for the importance of the underlying BIS technology. This can lead to technology (and standardisation) that facilitates very advanced integration of the logistics chain, both upstream (the supplier's side) and downstream (to customers). See, for example, efforts to achieve supply chain management. The result is that the BIS development is driven increasingly by technology.

2107 A development and control cycle lasting as long as ten years, as mentioned before, is questionable. Instead of this one could wind up in a continuous cycle of renewal (not only technological). For the BIS this means that one would be moving along with all supplier and system developments, by implementing new releases and successive versions of systems. Here, the aim would be to continually have a modern BIS at one's disposal as a weapon against the competition. Essential criteria in the selection of the BIS therefore include the continuity and innovative capacity of the supplier.

2108 One important warning should be made concerning the strategic role of information technology certainly where it concerns the use of standard available components and systems, namely competitors will also be able to acquire these systems and components relatively easily. The essence of this approach is then to strive for continuous improvement.

Organisational size

2109 It is obvious that the size (and therefore the financial power) of an enterprise plays an important role in determining the starting point for the selection of a BIS. One could distinguish three types each with a different approach to the selection process. As the approach seems to reflect the size of organisations, this size is used to characterise these typical approaches. For that matter, one can only speak of a general characterisation and not of a scientifically crystallised typology. The following three extremes can be distinguished:-

(a) *large and complex organisations* with a complex multi-national organisational structure and a hybrid type of company. For example, there is a multi-national sales and marketing organisation (often in a business unit structure) next to a diversified production organisation, and furthermore the distribution and transport are so large that it is directly controlled by the company and is therefore part of the organisational framework;

(b) *medium-sized organisations* where the organisational structure provides the focus on a specific core-business. For instance, linked departments for sales, purchase, production and financials;

(c) *small organisations* that exists due to their limited (but specialised) core-business and of which the organisational structure and division of labour is relatively flexible. The activities of a number of supporting functions are (or can be) custom-made and all expenditure is directly aimed at short-term (financial) return.

2110 One should add here that only a rough characterisation is given and there are probably more mixtures in actual practice than organisations that exactly fit these caricatures. This classification is used to relate to the characteristics of information systems, selection processes, and relevant selection criteria.

Large and complex organisations

2111 Possible characteristics of large and complex organisations in relation to information systems are:-

(a) a real-time interface between modules (for example between manufacturing and financials) is often less important and only a (batch) financial consolidation is required. Since there is a business unit organisational structure, batch interfaces between systems are often more practical. Thus one could choose different systems and suppliers for the various business areas for example, three different systems for: financials for the holding and the business units; personnel management for the entire enterprise, and manufacturing control for one production unit;

(b) the in-house information management department is capable of gaining detailed knowledge of all systems in question, and can therefore carry out a substantial part of implementation, maintenance and management activities under their own steam.

2112 For the selection process this can imply:-

(a) the choice of the market leaders, for example only the top 5;

(b) the top 5 is defined as the largest suppliers, based on their international presence, support, and a broad spectrum of modules;

(c) the functional details offered in specific modules is subordinate;

(d) the selection process is mainly focused on negotiations with the supplier.

2113 Important considerations in making a choice are:-

(a) the price of the complete package for large numbers of users and installations;

(b) the international support;

(c) the continuity of the supplier in concerning future developments.

2114 The validity and further discussion of these characteristics falls outside this scope.

Medium-sized organisations

2115 Possible characteristics of medium-sized organisations in relation to information systems are:-

(a) the integration of functionality and modules is extremely important. Due to the close co-operation between departments and business functions, the strength of such organisations lies in being able to respond quickly to demands and developments in the market. Thus, the complete integration between sales, production, purchase, and financials make a business process-oriented working practice possible (as opposed to a functionally organised working practice);
(b) the limited IT staff focuses mainly on systems management, first line user support, and is the central contact for all software and hardware suppliers.

2116 For the selection process this has the following consequences:-
(a) the logistics and economics control structure is an essential basis for the selection;
(b) the detailed functionality offered is an important selection criterion;
(c) all required modules or functions should be available from the supplier, i.e. besides purchasing, manufacturing, and financials, maintenance and personnel management should also be available;
(d) the selection process is mainly aimed at the evaluation of the functionality provided;
(e) the involvement of the key users is of great importance for contributing business knowledge and for creating a base for acceptance.

2117 Important considerations in making a choice are:-
(a) the suitability of the functionality;
(b) providing a total solution, including hardware, infrastructure, software, and support. This is also referred to as one-stop shopping;
(c) the suitability of the supplier as a partner during and following the implementation;
(d) the continuity of the supplier.

Small organisations

2118 Possible characteristics of smaller organisations in relation to information systems are:-
(a) the information system is viewed as a loose collection of tools that is only aimed at the direct support of the business activities. There is only a need for functional integration of information process concerning the primary processes (for example, raw material stock administration interfaces to the purchasing module, and integration of the invoicing module and the financial administration);
(b) the information system is only regarded as a tool for automating manual activities;
(c) there is no computer department, but just a single specialist who performs the daily tasks of a system manager as well as some other (line) function (usually in the financial department).

2119 For the selection process this implies:-
(a) a choice for the simplest and cheapest systems;
(b) a choice based mostly on the presence of the required modules and functions;
(c) the detailed functionality has to be sufficient, but detailed requirements are very limited;
(d) the selection process is aimed at cost efficiency.

2120 Important considerations in making a choice are:-
(a) offering a total solution, including hardware, infrastructure, software, and support;
(b) the price of the total system (hardware, software, infrastructure and support);
(c) the support provided by the supplier locally;
(d) the continuity of the supplier.

2121 The detailed study of functionality within the framework of this research is particularly relevant for the (largest) group, the medium-sized organisations. For the large and complex multi-nationals, the functionality offered is relevant, but only comes into play at a later stage than the actual purchase, i.e. during implementation and training.

Availability of Business Information Systems

2122 A sufficient supply of standard software is of course a condition for starting a software selection project. The number of potential candidates does not need to be large since only one system is required, but the few candidates considered must have a reasonable chance in order to participate in the selection process.

2123 For many kinds of organisations and business processes, a quantitatively sufficient supply of standard BIS is no problem. Thus, there are hundreds of standard systems available for normal manufacturing environments. However, there is a number of branches where the supply is far more limited, for example the textile industries, the printing industries and even transportation companies. This limited supply is generally caused by:-

(a) the specific characteristics of the product or the manufacturing process (like in the textile branch);
(b) the small scale of companies in an industry sector (as in road transport);
(c) the limited number of very large companies that are completely self-sufficient (as in railway exploitation).

2124 A limited number of BIS is also offered in branches where information processing itself is the primary process (instead of being a secondary process), as is the case for insurance companies, banks, and leasing companies. An explanation for this could be that the information system is exactly distinguishing an organisation from its competitors. Purchasing a standard BIS that is also used by the competition is only acceptable for (smaller) organisations that excel themselves in other areas, like personal customer relations and flexibility.

2125 Depending on the supply of standard BIS, the following aspects should be considered (preferably in the order given):-

(a) additional modifications and developments by the supplier or by an in-house IT department;
(b) in-house development on the basis of the available templates or semi-developed software modules;
(c) the full development of a BIS from scratch.

2126 Certainly in the last two cases, additional requirements have to be made of the development tools chosen and the underlying technology. Basically the technology, and the availability of knowledge and experience using it, is of major importance in the selection of available templates.

2127 In a selection process, many activities have to be performed that would also take place if one was to fully develop a BIS. But if this is the objective from the start, another approach (focusing on information systems development) would be more appropriate. Within this context, the assumption is made that a sufficient supply of standard BIS is available. However some modifications may still be required.

Internal policies on information strategy

2128 Finally, an overview is given of more internally focused aspects that influence the selection process as part of the information management strategy. The most important considerations are:-

(a) the constraints concerning the *hardware platform and the technical infrastructure*. Sometimes usable hardware is available, or more importantly valuable expertise has been built up around a certain technology. Such constraints can also be placed upon the external data communication, like EDI standards or use of Internet;

(b) the requirements regarding *development tools*, such as database management systems, case tools, programming languages, query and reporting tools etc. The most important reasons to determine such requirements are the tools and expertise already available, as well as the interfacing or integration with existing software systems;

(c) the strategy regarding *software modification* on the BIS. Earlier some specific situations have been mentioned in which modifications are unavoidable. However, one has to realise that at best a fit of about 90% can be reached with a standard BIS. Whether or not one should achieve the remaining 10% (or more) by modifying the BIS is very debatable. Two strategies are possible:-
    (i) principally no software modifications to be made unless it is on the outside of the system (for example report functions) so new releases can be installed without any software conversion;
    (ii) allow for software modifications only if the costs incurred are less then the alternative of adapting the working practice or executing activities manually; (note: a third strategy in which software modifications are realised without any visible return is not considered useful at all);

(d) the policy regarding *new releases*. This point is related to the previous point on modifications. Again two principle policies are possible:-
    (i) only choose to implement new releases only if they cover an existing technical or functional requirement, where explicit cost-benefit calculations have to be made for each upgrade (however if previous releases where not implemented, here also this backlog has to be taken into account);
    (ii) always choose to implement new releases to keep the BIS fully up to date and to have a modern information system continually at one's disposal;

(e) the integration with other types of *office automation*. Next to an integrated BIS, every organisation uses additional software for word processing, supplementary reporting, spread-sheet and decision calculations, diary management and ad hoc personal systems. The degree to which these local applications need to be integrated is a strategic choice. Technically, modern BIS can go quite far in integrating office applications, but they do make demands on certain standards;

(f) *security aspects*. Here, it concerns the requirements and procedures for back-ups, recovery, an alternative computer centre, and of course the authorisation of users. As regards the latter, one should also take external users of the BIS into account. Consider for example (future) situations in which customers directly check their own address data in the BIS, or place orders, or can look at the current status of the stock or deliveries (maybe using Internet).

2129 This brings close the most important aspects concerning information management strategy, with a particular focus on the selection of a standard BIS.

Best practice experience in software selection

2200 Before one can start with the selection of a complex BIS, one needs to determine in which direction the organisation, company management and control is going to develop. Firstly the logistics, commercial and financial (controlling) concept is considered, which will hereafter be abbreviated to 'business concept'. Secondly, a broadly accepted method for the selection of software is described. It consists of roughly four phases each containing a series of compulsory and optional activities. Following that, the critical success factors for the selection process are detailed and the establishment of a project organisation is considered. Finally, a list is given of the most important areas of expertise required for an effective selection. Here, the strengths and weaknesses of the potential providers of this expertise are compared.

Starting point is a business concept

2201 The (traditional) statement of requirements with respect to a new BIS is typically determined by the users of the system. The management tends to focus on the existing situation while closing up the requirements and mainly considers the shortcomings of the current information system.

2202 Frequently lacking in this traditional approach is:-
(a) the distinction of essential versus nice-to-have requirements that may save labour but that are not crucial to the performance and control of the primary processes;
(b) the introduction of future strategic developments.

2203 As a result, the new BIS may meet 80% of the requirements stated, but does not fit the fundamental (logistics) characteristics of the primary processes and their control, nor will it fit after a strategic reorientation with regard to the logistics or commercial processes.

2204 A logistics, commercial and financial (controlling) concept (the business concept) is the foundation for proper management. Though this business concept is in many cases not described in detail, employees in an organisation are aware of it, at least the parts concerning their own function. It is the guiding principle for their daily activities. As a starting point in a process of change, which certainly the implementation of a new BIS is, it is worth describing the present business concept explicitly.

2205 In the process of change, the selection and implementation of a new BIS, such a business concept is essential for the future management of the business. It focuses the change process and brings to light the essential functionality critical for the realisation of that business concept. In practice, the implementation of a new BIS is also regarded as an opportunity to implement more changes than just the new BIS, i.e. the further optimisation of the business concept.

2206 Improvements being indicated and realised can be distinguished into two groups:-
(a) purely logistics, organisational, or process-oriented improvements that are (or can be) independent from the introduction of a BIS. Take for example:-
  (i) the alteration of the lay-out in manufacturing;
  (ii) the formation of a logistics department;
  (iii) the simplification of commercial (pricing) agreements with customers;
(b) improvements that are possible through the use of information technology in general and the implementation of a new BIS in particular. Here, we are referring to:-
  (i) the elimination of duplicated activities or the computerisation of manual activities;

(ii) providing extra information to customers to enhance the added value or to improve customer relations;
(iii) reducing the stock levels by being capable of making sophisticated forecasts based on of detailed historical data.

2207 For the realisation of this first group of improvements it can be argued that they can better not be linked to the introduction of a new BIS. The degree to which an organisation can undergo change is limited, the daily work should be able to continue. This means that such improvements either have to be realised before a new BIS is introduced, or that they should be tackled when the new BIS has been operational for some time.

2208 Considering the importance of the business concept in this context, we give a brief elaboration is given of its most important elements. For a more extensive description the reader is referred to the literature in this area (for example [BER90] and [C&L91]). The business concept can generally be determined in the following areas:-

(a) *the market areas, the product assortment and thus the characteristics of the product-market combinations.*
This typology has to be elaborated for aspects like the development of turnover, market demands, features compared to the competition, characteristics of the product, product structure, and of course mutual interaction of all these aspects;
(b) *the characteristic description of the primary processes.*
In a manufacturing or distribution environment this is also referred to as the logistics structure. Here, the following is described schematically:-
    (i) the steps that are distinguished in the primary process, such as the (manufacturing) operations, the design process and the transport;
    (ii) the buffers and stock points that can be found in the logistics chain;
    (iii) the customer order decoupling point indicating at which points the customer order can be recognised in the primary process;
    (iv) a quantification of the processing times in relation to the through-put times;
(c) *the planning and controlling structure*, that describes how the primary process is managed.
Some important aspects are: the forecasting process by marketing and sales, the criteria for accepting customer orders, the horizon, frequency, levels and units used to planning, and the organisational responsibilities;
(d) *the logistics, commercial and economical costs, index figures and targets.* Together they form the foundation for the deduction of performance indicators (as index figures that can be influenced and with which the company can actually be controlled).
Examples of (operational) costs are: purchase costs, storage and handling costs, transporting costs, and marketing costs. Other index figures include the stock value, the through-put times, the capacity utilisation, the delivery times.

2209 The summary above is not intended to be complete. It only serves to provide an adequate framework for the rest of this chapter.

2210 In order to achieve improvements in all these areas, one can make use of a large number of strategies and techniques. Thus, logistic strategies aim at the reduction of the (technical) variability, for example, the prevention of malfunctions through preventative maintenance or the standardisation of components used. In other areas an attempt is made to increase flexibility, for example concerning the (multiple) deployment of resources.

2211 For a process-orientated approach in the search for improvement potential, techniques are borrowed from Business Process Re-engineering (BPR). This approach is aimed at elimination, reduction, combination, and (finally) computerisation of activities.

2212 The most important aspect of these approaches is the creativity required to achieve revolutionary improvements, the innovation may also be supported by opportunities provided by information technology and modern BIS.

Best practice selection method

2213 Earlier we indicated how an optimal starting point appears before starting out on the selection of a BIS: the guidelines with respect to the information management strategy and a (brief) description of the business concept to be supported by the BIS.

2214 The selection process that follows should provide us with more than just a suitable BIS. Some good objectives for this process are:-

(a) the choice of a *sufficiently suitable* BIS;

(b) the choice of the *suitable suppliers* for software, hardware, technical infrastructure, training and all other required components and services. Naturally, this all depends on the areas in which one is self-supporting, including relevant experience and expertise. The internal computer centre can also be considered here as a supplier;

(c) the building up sufficient *knowledge to plan* an achievable implementation (or at least to be able to assess this plan) and the ability to examine the consequences of this for the organisation;

(d) the formulation of a rough *plan for the implementation*. The minimum requirements are the phasing and timing (also in connection with the delivery of goods and services); the project organisation with its tasks and responsibilities (and with that those of the supplier(s)); and preferably a more detailed activity plan;

(e) the creation of a *support base in the organisation* for the selection, the implementation and the organisational consequences;

(f) the conclusion of an *optimal contract* with the supplier(s). All relevant elements have to be appended to this contract, thus also the implementation plan, the functionality to be delivered and the mutual responsibilities (note that this list is far from complete).

2215 In essence the selection process is the step by step reduction of the possibilities for choice. In other words, at the beginning of the selection process one can choose from the entire spectrum of BIS that are available on the market, and in the end this choice should have been narrowed down to exactly one option. Considering the breadth of the supply of potential candidates (at least for most branches), it is obvious that this process of choosing is taken in a number of steps. The wrong approach is to choose by means of a very long questionnaire or an other detailed form of evaluation, where one single candidate is selected from all possible candidates in a single step. For each step a minimum of activities should be performed to realise the necessary reduction, so detailed evaluation should only take place for a limited number of candidate BIS.

2216 On the other hand, it could be the case, normally as a result of the information management strategy developed, that a principle choice is made in advance for one market leader. This does not mean, however, that the selection process is finished, consider all the other objectives mentioned earlier. The effect is just a simplification of this process, which results in an evaluation of this principle choice (this is often referred to as a feasibility study). This evaluation still is essential for the functional assessment (what is the actual functionality of the product being purchased), the formation of an implementation plan (what can the organisation expect from this functionality and what will be the consequences), an estimation of modifications (what is the mismatch between offered functionality and the requirements), these aspects outline the candidates for being able to enter into an optimal contract.

2217 In figure 2.1 the possible steps are presented for this structured reduction of the possible choices. It should be noted that some tasks are optional depending on the starting point.

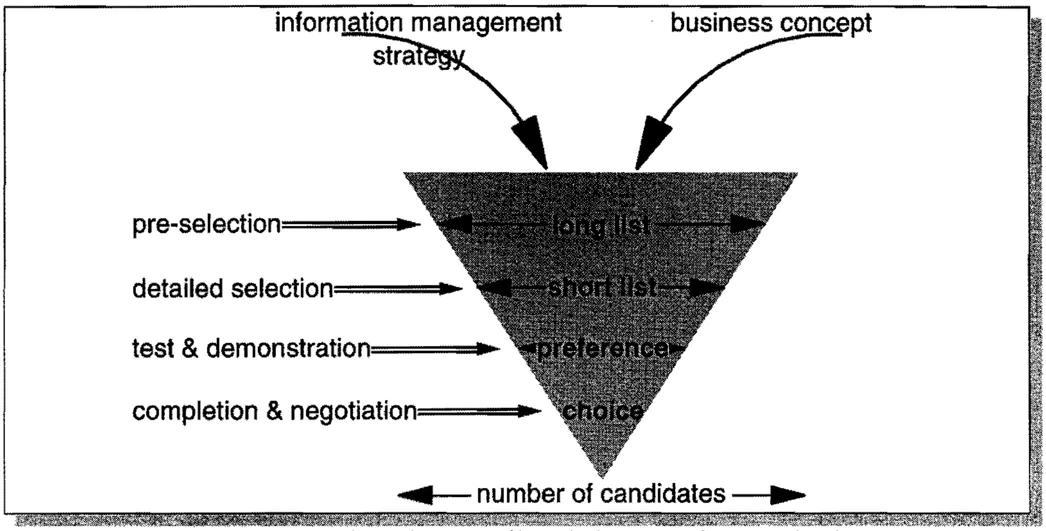


Figure 2.1 - Step by step reduction of the possible choices in software selection

2218 Here these phases will successively be detailed and discussed. However, no attempt will be made to give a complete route-map or checklist for software selection, with which an organisation could do a step by step selection. The objectives, approach, and essential characteristics of each step will be discussed. An extensive summary of considerations, pitfalls, and practical examples is beyond the scope of this dissertation. A number of variations that can occur in the various phases will be noted. There are always small differences in approach and especially in the tools that are used. This is partially due to the knowledge and experience of the parties involved. With the exception of relatively small differences, there is broad acceptance of this method for software selection and particularly among consultancy firms.

Pre-selection

2219 The pre-selection is aimed at mapping the supplier's market of possibly relevant BIS. It should result in a long list of candidate BIS and suppliers. In practice it is feasible to restrict the long list to fewer than ten candidates.

2220 At the start of the pre-selection the following should be available:-
(a) the information management strategy;
(b) the logistics, commercial and economic business concept;
(c) the general branch-oriented characteristics.

2221 On the basis of research, one can draw on a vast amount of material and documentation, since all these suppliers have to stress their distinctive features and are very active in spreading information about their product assortment. Important sources for the pre-selection are the general documentation of systems, business fairs, branch organisations, experiences of similar companies, evaluations in professional magazines, supplier's advertisements, published comparisons of BIS, and expert's experiences. The art is to restrict the multitude of suppliers to fewer than ten. If this is problematic, it is best to focus on real practical (proven) experience with the BIS offered and with the suppliers.

2222 The pre-selection can be skipped as a separate phase if one is familiar with the market. Basically, this step will be executed implicitly when drawing up the short list.

Detailed selection

2223 The purpose of the detailed selection is to draw up a short list of three suitable BIS, as well as the (accompanying) suppliers who are capable of supporting the complete implementation. Besides this short list, the results of this phase also consist of:-
(a) *the information concerning the suitability* of the BIS as regards the critical assessment criteria;
(b) *the supplementary information about the BIS* concerning the history, the number of implementations and references, the future plans and expectations, and the possible hardware and system software;
(c) *the supplementary information concerning the supplier's organisation*;
(d) *an indication of costs* for various components and services.

2224 The most important element in the detailed selection approach are the '*knock-out criteria*' or '*important requirements*' (depending on which philosophy one follows). These criteria have to be met by the potential candidates to be part of the short list. In principle these criteria are used on the candidates from the long list, after which the most suitable ones remain.

2225 *If the detailed selection is based on 'knock-out criteria'*, about twenty discriminating criteria are collected. These criteria are so critical that the only BIS (and suppliers) that meet all these criteria will be put on the short list. In this approach, a number of additional requirements are sometimes attached to gain some more functional information.

2226 *If the detailed selection is based on 'important requirements'*, no more than about fifty (discriminating) requirements should be collected (though they should of course be the most important ones). A larger number of requirements quickly leads to a distortion in the process of choosing. By scoring on these 'important requirements' the short list can be composed. In contrast to the approach based on 'knock-out criteria', the second approach also provides a list of BIS in order of suitability (thus can later be used to determine the sequence of the demonstrations).

2227 To test both kinds of criteria, one can make use of existing databases and comparisons, which are usually owned by specialised consultancy firms. This only applies to the extent that the criteria are covered by these databases and also that the data on the BIS is up to date (considering the rapid developments). Alternatives are to directly gain information from the suppliers, to study the documentation, and to have the suppliers fill out a short written questionnaire.

2228 In any case the short listed suppliers have to confirm in writing whether or not and to what extent the formulated criteria can be met. Here, it is also possible to inquire about supplementary information regarding the BIS, the supplier organisation, and costs (see the objectives mentioned earlier). It is also advisable when one first becomes acquainted with the (for example three) candidates that they present themselves and their BIS to the project team and respond to some additional questions.

2229 If a principle choice for a BIS has been made in advance and can no longer be influenced, a number of the activities mentioned should still be carried out with that one candidate, to gather the missing information. This is an important step in creating commitment from the supplier's organisation and a support base in one's own organisation.

Test & demonstration

2230 The tests and demonstrations are meant to assess the candidates on the short list on important aspects. This should result in a sequence that makes the principle choice possible. In other words, a particular candidate will be chosen unless inaccuracies are uncovered in the next phase or the results of negotiations are negative. Besides this sequence, a number of other results should be provided in this phase:-

(a) quantitative analysis of the most important database and transaction volumes;
(b) summary of the required functions per module;
(c) global scenario for the hardware and technical infrastructure based on the number of users and the physical situation;
(d) cost indication based on the above for all required components and services.

2231 Based on a specific business case, consisting of realistic data and business processes, all (for example three) candidates have to give a comparable demonstration. In this the supplier has to simulate the critical processes of the company. Where drastic improvements in the working practice are proposed, it is vital that the demonstrations be based on the new and not on the current working practices.

2232 Besides the business case being used for the demonstration in this phase, it can be used in later stages as a starting point for a more extensive workshop, as well as for training purposes and for configuring and testing the finally selected system.

2233 As was mentioned in the previous phase, a more extensive list (with a maximum of fifty) of important requirements will also give a sequence of the short listed candidates. In practice, the case demonstration is only carried out for best candidate. If this candidate does not provide the functionality expected then one can always turn to the other candidates. A strong disadvantage is that comparison is not possible and that no other case demonstrations are available as a reference.

Completion & negotiation

2234 The objective of the last phase is to verify the principle choice and to make all final the decisions concerning the completion of the selection process. This phase should result in:-

(a) a (functional) confirmation of the (in principle) chosen candidate from the short list, as far as there was any doubt left of to the previous phase;
(b) an impression of the software modifications required and the consequences for the organisation;
(c) an implementation plan based on the above, including the responsibilities and tasks (and those of the supplier), the phasing and timing of all activities and the project organisation;
(d) a contractual agreement composed of all of the elements listed above;
(e) the good support from within the organisation for the contracted BIS as well as for the implementation plan.

2235 To verify the principle choice, references should be checked. In addition, a workshop is sometimes organised that, besides the verification also gives more information for making the implementation plan. The objective is twofold: a detailed verification of the functionality across the full scope and the further development of the implementation plan. Such a workshop is also referred to as a Conference Room Pilot (CRP). This name is also used for the first stage of a method for implementing BIS, this approach is indeed comparable to this kind of workshop. Note, however, that the objectives of these two kinds of CRP's differ (verification versus implementation).

2236 The preparation of the workshop consists of a further expansion of the business case to regular processes and the training of a small group of key users. Next, a number of weeks are set aside to test the entire BIS in co-operation with the supplier. This gives the key users involved a good idea of the possibilities offered by the BIS and the work required to be carried out in the implementation. The (usually paid) supplier's involvement also gives a good indication of the future co-operation.

2237 Nevertheless in many cases this workshop does not take place because:-
(a) a number of complex BIS cannot be configured in so short a time. On the other hand, if this really is the case then the business case demonstration can not provide adequate insight into the functionality offered!
(b) one is eager to start with the actual implementation. However, this workshop makes a few initial activities superfluous, so that one can get off to a flying start;
(c) the required intensive efforts, during a couple of weeks, of the key users cannot be organised. On the other hand, one should consider that during an efficient implementation a considerable amount of time will be required from the same key users anyway.

2238 An important advantage of the workshop is the direct involvement in and the commitment of the supplier, who has to continually prove the performance of the software and the organisation before a final contract is agreed.

2239 If we assume a positive result for the workshop (if it is negative this would indicate that earlier parts of the selection where not carried out properly), the last step involves the negotiations with the BIS supplier and possible other parties who are to provide supplementary components or services. Besides the financial aspects, we not only need the complete specification of the software, hardware, and technical infrastructure, but also elements like training, support, maintenance and new releases. By making the right preparations in the previous phases (noting promises made well in advance), the last step will lead to an optimal contract.

2240 If the software supplier has a more subordinate role in the selection process, because there is for example, an in-house computer department, then the activities described above will shift towards actual functional aspects. Elements like the supplier's commitment, support, prices, and contracts are often less import in such situations.

2241 A deviation from the prescribed selection method can be acceptable for various reasons:-
(a) the evaluation of one BIS (feasibility study) instead of a selection process;
(b) the compulsory use of an in-house computer department.

2242 However, the quality of the final result (the right choice among other things) is hard to measure and is usually not measured at all. This means that mistakes in the selection process are only found out at a very late stage or remain undetected. A system is chosen and a contract is concluded but the mistakes made (or opportunities missed) in the earlier stages only become clear during the implementation or when the system is operational. Often they never become visible, but the result is an overly expensive or delayed result.

Critical success factors and the project organisation

2243 Earlier we described the phases in the selection process with a particular focus on the objectives. A number of critical elements function as a guideline through the phased activities in order to actually achieve these objectives. In short, the objectives of the selection process are the following (see paragraph 2214):-

(a) the choice of a sufficiently suitable BIS;
(b) the choice of a suitable supplier;
(c) the expertise to carry out a successful implementation process;
(d) a rough implementation plan;
(e) the general support from within the organisation (users and management) for the selected BIS;
(f) an optimal contract with the supplier(s).

2244 To achieve these objectives we can identify a number of critical success factors:-

(a) *the right functional assessment criteria*, which are based on the characteristics of the business concept (and aimed at **objective (a)**). If these essential criteria are not identified, one cannot guarantee that the right BIS will be chosen (or that a choice is made for a BIS that is far too large and too extensive);
(b) *the starting points concerning the information management strategy* (aimed at **objectives (b) and (f)**). The product one purchases encompasses far more than the single delivery of software. For the long run one should consider the supplementary services, the continuity of the IT architectures and organisations, and the future role of and co-operation with the supplier(s);
(c) *a sound test of the essential starting point and criteria* (aimed at **objectives (a), (b), and (c)**). It still happens that companies fall for the commercial talk or beautiful presentations of a supplier. The result is a chance hit or a bad buy. This could mean starting the implementation process without being fully aware of the consequences;
(d) *a realistic determination of the benefits of the integral process of change* (aimed at **objective (e)**). It is import to the entire organisation and especially to management that benefits be clearly visible. This is the best way to deal with the costs, the time required, the efforts, and the inevitable resistance to the fundamental process of change. These benefits are generally the result of the identified improvements that were found while the business concept was being drawn up;
(e) *the availability of sufficient expertise and experience in all relevant areas* (aimed at **all objectives**). This expertise in the selection process is further elaborated upon from paragraph 2249 onwards;
(f) *the commitment of the management, the users, and the suppliers* (aimed at **objective (e)**). The selection process is a lot more than just a rational decision hierarchy. All the parties involved have to be really dedicated to the selection process and of course the implementation). True commitment is expressed among other things in interest in and attention for the content of the activities that one is *not* responsible for. It regularly occurs that the commitment from the supplier is lacking, this is an inevitable result of involving too many suppliers in the selection and giving them too much detailed work to do (usually in the shape of long questionnaires). For a supplier the chances of success are far too small compared to the effort he has to put in. A customer-supplier relation is not one way as it is traditionally believed to be, and certainly not when entering into a complex and sizeable implementation process together. The essence of the 'best practice' approach as sketched above is that the effort one puts in has to be inversely proportional to the number of candidates;
(g) *an effective project organisation and division of tasks* (aimed at **objective (e)**).

2245 The selection process and the implementation in particular is far more than a number of rational choices made step by step in a series of activities. The activities are complex by nature, and this is definitely true for a user organisation that is focused on tasks other than the selection and implementation of a BIS. Therefore a project organisation including task division is required that:-

(a) focuses on the project objectives, and particularly gaining on a general support among all parties involved;

(b) is a fit with the knowledge and experience in all relevant areas of the organisation;

(c) effectively controls the vast number of activities within a given financial budget and time scale with the necessary quality;

(d) capable of realising all activities effectively.

2246 In the area of project organisations, management, and planning, a lot of information is available. It is beyond the scope of this investigation, but readers are referred to the literature, for example [C&L87].

2247 There are a few specific considerations, specially formulated for this type of project:-

(a) the formal leadership and responsibility for the project should (preferably) be in the hands of the line management, and as close as possible to the future owners of the BIS. In other words, it should not lie with the traditional IT department or any other staff function, and certainly not be in the hands of any external consultant;

(b) the project leader should continually report back to a steering committee at a management level covering the entire organisation (and therefore the entire scope of the company-wide BIS). All important decisions that have been or have to be made, should be confirmed here;

(c) within the scope of the BIS, all (important) functions and departments should be represented. This should preferably be done by the heads of departments or delegated (but fully authorised) key users. It is not only essential that functional knowledge is present in the project, but also that (fundamental) choices can be made quickly. During the execution of the activities, the responsible parties are of course free to delegate work, but they should maintain full understanding of the content and should monitor and control the activities;

(d) during the execution of activities, one should realise that the knowledge and experience of key users does not normally fit this kind of work. This can be compensated to some extent through specific training, for example in the area of logistics. Nevertheless, in many cases this has been a reason to *support* the participants with 'business templates' or by 'experts'. The same can apply to the project leader;

(e) besides the knowledge mentioned, the required capacity of the project team members has to be taken into account too. For the duration of the project, the effort to be invested is high, at least thirty or forty per cent of the job time. This means that special measures have to be taken regarding the normal (daily) work;

(f) in the division of labour within the selection process there are different strategies:-

    (i) traditionally, organisations often work with an expert on the project. These internal or external experts hold the interviews, presentations, report progress and hold formal feedback sessions. The role of the expert can be fulfilled by system managers, information managers, heads of computer departments on specialised consultants;

    (ii) a more process-oriented division of labour, based on interactive workshops, which are led by an expert but where all participants play an equal role as project members;

(g) finally, the best results seem to be achieved by working on short time scales instead of striving to perfection. This causes a high workload (for a shorter period) but at the same time there is also a great deal of motivation.

2248 An effective project organisation proves to be the most critical success factor in practice, both regarding selection as well as implementation.

The role of an expert

2249 It is obvious that expertise plays an important role in selection processes. The required expertise lies in a number of essentially different areas, i.e.:-

(a) knowledge of the company and its business processes;

(b) knowledge of organisations and working practices in general, like in the area of logistics and finance, but also in control structures and in 'best practice' experiences;

(c) knowledge of the BIS supplier market and more particularly experiences with these systems and their suppliers, as well as knowledge of the functional and technical features of these BIS;

(d) knowledge in the area of information management both in a functional sense, for example the specification of requirements, the evaluation of information systems, and in a technical sense, for example client/server architectures, database management systems, and technical infrastructures;

(e) knowledge of selection processes with a special focus on pitfalls, risks, and critical success factors.

2250 Normally, the selection of a BIS for a company is a one-off activity and as a result there is limited knowledge in-house available. Table 2.1 shows the possible knowledge providers on the different areas mentioned.

knowledge areas		internal user organisation		external suppliers	
		normal line functions	specialist staff functions	suppliers BIS / IT	specialised consultants
(a)	internal user organisation	+	+/-	-	-
(b)	logistics / 'best practices'	-	+/-	+/-	+
(c)	supply market of BIS	-	+/-	-	+
(d)	information management	-	+	+	+
(e)	software selection process	-	+/-	-	+

Table 2.1 - Providers of expertise

2251 In table 2.1 it is clear that the company itself (**the internal user organisation**) naturally has the most significant knowledge about their business processes, the organisation and its characteristics. However, relevant knowledge is lacking for the other areas simply because software selection is not a regular activity. At best an employee may have gained supplementary knowledge from a former employment or through personal interest (which is by definition limited and therefore dangerous).

2252 Larger organisations often have their own computer departments or information management departments (**specialist staff functions**). The knowledge that is available differs greatly from one organisation to the next and depends partially on the background of the existing information systems (often no standard software). Based on internal experiences, these staff have a reasonable idea of their own organisation and business processes, though not as good as the real executives and those responsible in line functions.

2253 Knowledge of areas such as logistics or 'best practices' and the process of selection are often present to some extent, but are truly limited compared to specialist consultants.

2254 It is not feasible or useful to follow the BIS supplier market for the limited scope of one's own organisation. If this knowledge is present, it will be restricted to a few BIS and their suppliers, and restricted in detail.

2255 Expertise in the area of information management is often present as this is the core function of the department(s) or function(s) in question.

2256 The strengths of BIS suppliers and information technology in general (**suppliers of BIS/IT**) are of course in the area of information management. Extensive knowledge is available concerning the technical aspects. Depending on the supplier, knowledge and experience in areas like logistics and 'best practices' can sometimes be present too. However, this is not their core-business and therefore will not counterbalance specialised consultants.

2257 For the remaining aspects, detailed knowledge of the supplier market and experience with selection processes, the supplier is by definition biased. This means that one cannot and should not expect independent support to be provided by the suppliers in these areas. Even the fact that one is building up this knowledge is in conflict with the primary objective of the supplier company.

2258 Finally, there is a wide spread of consultancy firms (**specialised consultants**) that as a whole cover all these areas of knowledge well (with the exception of course of the specific knowledge of the particular company). Some of them take matters one step further and have a leading role in the developments in a specific area. These consultancy firms compete with the internal specialist staff functions (see table 2.1). Not every consultancy firm is equally qualified in all the areas mentioned. Thus, a distinction is made between technical consultants (often focused on information technology), business experts (aimed at areas like logistics and the BIS market) and management consultants. If one wishes to increase the quality of the knowledge that is already present internally, one will have to make a critical selection of the right consultancy firm (or combination of consultants).

2259 Besides experience with comparable selection processes, and branches or types of organisations, a number of specialised consultancy firms have a (huge) database with information about BIS and suppliers at their disposal. On the basis of the regular contacts with suppliers and comparable selections these databases are updated. So there are (internal) databases with many hundreds of criteria on which the important BIS can be graded, see for example [IPK94]. Besides such formalised checklists, factual experience and weaknesses are recorded into databases, see for example [C&L95].

2260 Sometimes those databases are published. Note however that these results (in the form of a book or on diskette) are static by definition, while the BIS market develops rapidly. This, may be a good instrument for the pre-selection and drawing up a short list, however these databases are not suitable for making the final choice.

Assessment of requirements

2300 In the assessment of BIS a multitude of requirements play a role. The most important group seems to be that of the functional criteria, though this need not always be the case. Excellent support from the supplier can easily solve a number of functional shortcomings in the system. In the following a typology of requirements is worked out. Then, the functional criteria are considered, on which this research focuses. Finally, the assessment of requirements and in particular the functional criteria is detailed.

Typology of requirements

2301 The requirements that can be listed when a BIS is to be purchased are numerous and very diverse in nature. Which of these requirements are truly important differs for each situation. On the basis of the requirements (and wishes) made, an assessment is made of the BIS as well as the suppliers. The results should be reflected in the contract as clearly as possible. Unfortunately, not all criteria are equally solid and therefore suitable for the contract, for instance the continuity of the supplier organisation.

2302 In chapter 1, from paragraph 1115 onwards, a very general division is given that is used by market researchers (in this case the Gartner Group) for the characterisation of the available BIS. This characterisation is focused, however, more on the trends and developments in the supply than on the support during the selection process. In the latter case we are interested in the specific selection criteria that differ from situation to situation, as well as their mutual degree of importance.

2303 A typology of assessment criteria is given below. Though they are not all equally important, at least not in every situation, one should make certain and base the assessment on the most important criteria. The typology consists of three main categories (supplier, system, and conditions), each subdivided into a number of subcategories. Per subcategory the most important examples of criteria have been listed for purposes of illustration.

2304 Criteria concerning the *supplier*:-

(a) *the structure of the present organisation*, such as the organisation structure and size (local / international), the dealer network, possible partners, and co-operatives;

(b) *the development of the organisation*, such as the history and growth in recent years, their view of the future, expectations for the future, and an estimation of their continuity;

(c) *the services offered by the organisation*. To be further subdivided:-

  (i) *nature*: such as what kind of support is provided: training, implementation support, software modifications and development, customisation, and consultancy;

  (ii) *reputation*: such as references, number of implementations, an opinion on the quality of services, and the supplier understanding of client situations;

(d) *the financial situation*: such as turnover, personal capital, liquidity, and investments that are made in research and development.

2305 Criteria concerning the *Business Information System*:-

(a) *the development of the system*, such as the history and expectations for future of the system, the planned developments both in a functional and technical sense, and the release policy;

(b) *the functionality*. This criteria is too large to be worked out here in detail. From paragraph 2309 on we shall give further details on the specification of functionality;

(c) *the technology*. This can be further subdivided into:-

(i) *hardware*: such as the platform (central and decentralised in for example a client/server architecture), peripheral equipment with a distinction between (character based) terminals, PCs and work stations, other peripheral equipment such as printers and back-up units, and the technical infrastructure such as the network and data communication;

(ii) *system software*: such as the database management system, the (graphical) user interface, the development tools used (self-developed versus independent tools), their nature (for example object-oriented) and possible links with CASE tools, possible operating systems, and the support of (certain types of) client/server architectures;

(iii) *application software*: such as the quality and the maintainability of the design, software and data structure, the user-oriented customisation facilities, such as parameter control or work-flow control (see also figure 2.4), the availability of the source code of the application, and the level of integration achievable with other systems;

(d) *the performance*: such as the user friendliness of the system, the response times in relation to the hardware and system software configuration, and the dialogue and menu structure;

(e) *the security*: such as the user authorisation facilities, back-up and recovery facilities, the transaction logging, the audit trail of data and transactions, and the general data integrity and the system performance;

(f) *the documentation*: such as the user manuals, but also the documentation for implementation, maintenance and management, the on-line help information and possible context dependency, the training material, special aspects such as the language in which the documentation material in question is available in, and the shape in which it is available (on paper, on-line, via multi-media);

(g) *the implementation*: such as the number of implementations for example per industry sector or manufacturing controlling structure, the references for the current and the previous releases of the BIS, the complexity of the implementation, and the tools that are available for supporting the implementation.

2306 Criteria concerning the *conditions of delivery*:-

(a) *the investment,* the price of all components: including the hardware, the system software, and the application software;

(b) *the implementation costs*: including training costs, customisation, modifications, the installation and configuration of the BIS, and the general support by the supplier;

(c) *maintenance and support costs*: including maintenance fee to all purchased components, the purchase and installation of new releases, the use of the supplier's help desk, and possible periodic check-ups;

(d) *general conditions:* including the warranty provisions, the maximum delivery time of all goods and services, the response time for malfunctions or problems, secrecy, payment instalments, the acceptance procedure for goods and services and the transfer of ownership and risk.

2307 For a number of these assessment criteria, information can be provided by the supplier. However, many criteria are subjective and will have to be collected during all the dealings with the supplier, such as demonstrations, presentations, and negotiations. The opinion of (independent) third parties regarding these aspects is interesting too, consult experienced consultants, supplier relations, and existing (or old) customers.

2308 It should be clear that considering the large number of possible requirements, it is important to make priorities without overlooking essential issues.

Functional specification and requirements

2309 An important aspect on which BIS are assessed is the functionality. The functional criteria for an organisation stem from the design of the desired situation, these criteria can be deduced from the business concept and the step by step detailing that follows.

2310 Next to this, the further (functional) elaboration is one of the most important activities in the implementation. Here, this functional specification is focused on the (further) specification of software modifications and possible development of additional program functions and modules. Also in the customisation of BIS, that currently have a large number of degrees of freedom, the required functionality is configured for instance by setting parameters. Furthermore, during the implementation the process descriptions, procedures, and working instructions will be developed, which also are a form of detailing functionality.

2311 In the selection process the functional specification is aimed at:-

(a) the design of the business concept (focused on the physical structures and the decision structures);

(b) the design of the process improvements (focused on the flow and timing of data and activities);

(c) the specification of (essential) requirements;

(d) the description of the business case;

(e) the elaboration of the workshop construction.

2312 From the area of systems development, a large number of methods and techniques has arisen for the specification of functionality. Examples of graphical techniques that originated in this are:-

(a) Data Flow Diagrams (DFD), in which processes and data flows between these processes are modelled in a hierarchical way [DEM79]. This technique is especially suitable for the **activities (b), (d), and (e)** listed above;

(b) Information Systems work and Analysis of Changes (ISAC), where the hierarchical 'information precedence diagrams' (I schemes) are developed on the basis of an 'activity diagram' (A scheme) [LUN81]. This technique is mainly suitable for the **activities (b), (d), and (e)** listed above.

2313 Basically, the techniques originating from the area of systems development are mainly suitable for the support of the activities (b), (d), and (e) listed above.

2314 These techniques have gained many extensions and successors. One of these techniques i.e. the activity modelling from the Structured Analysis and Design Technique (SADT), see among others [ROS77], was adopted by the United States Air Force during the late seventies to support the Integrated Computer Aided Manufacturing (ICAM) programme. On the basis of this the ICAM Definition Methodology (IDEF0) was developed; a method that has a large following. Though IDEF0 has far broader applications, the development and use was mainly intended for modelling manufacturing processes and their control structure. This makes IDEF0 also suitable for **activities (a), as well as (b), (d) and (e)** as listed above. Some additional description regarding IDEF0 is shown in figure 2.2 (for further reading see [USA81]).

2315 In the mean time IDEF0 has been extended with a number of additional techniques for information analysis (IDEF1), the design of relational databases (IDEF1X), dynamic analysis (IDEF2), process flow modelling (IDEF3), object-oriented data modelling (IDEF4), and a repository for very large projects (IDEF5).

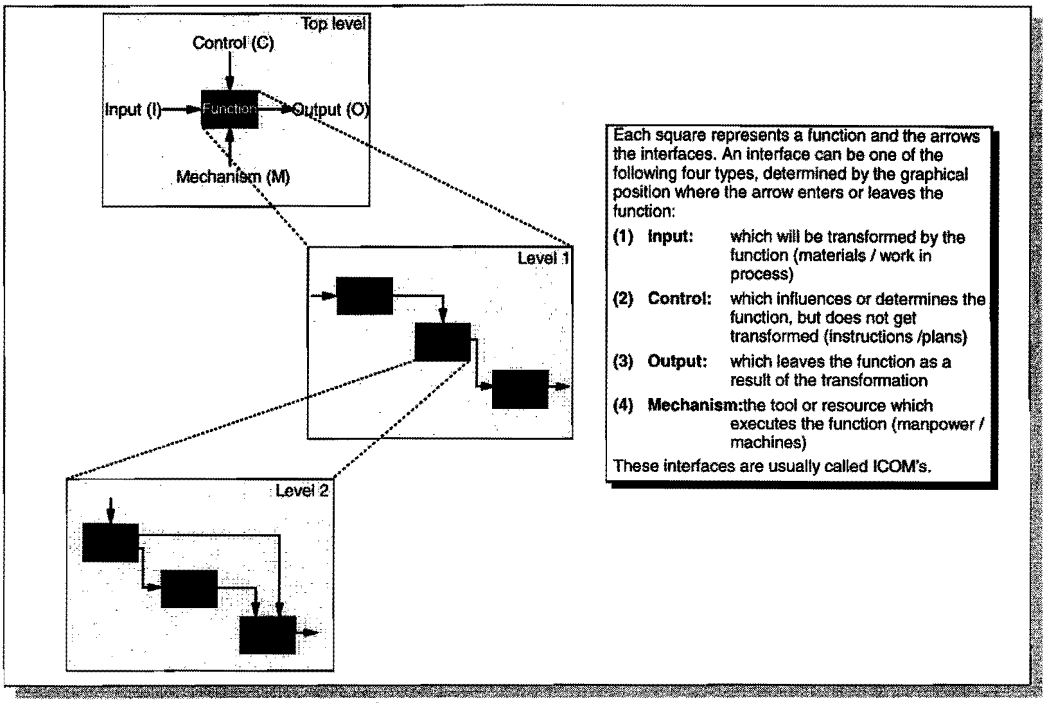


Figure 2.2 - Explanation of IDEF0

2316 Another method is Graphe à Resultats et Activities Interliés (GRAI), which was developed in the seventies by the French research centre with the same name. It focuses on the analysis and the design of manufacturing management systems. This method makes an explicit distinction between the physical system, the controlling structure, and the information system.

2317 Two graphical techniques are part of this method: the GRAI grid and the GRAI network. Specifically the first technique is often used for the modelling of control structures. As such it is particularly suitable for **activity (a)** as listed before. Figure 2.3 shows an example of a GRAI grid.

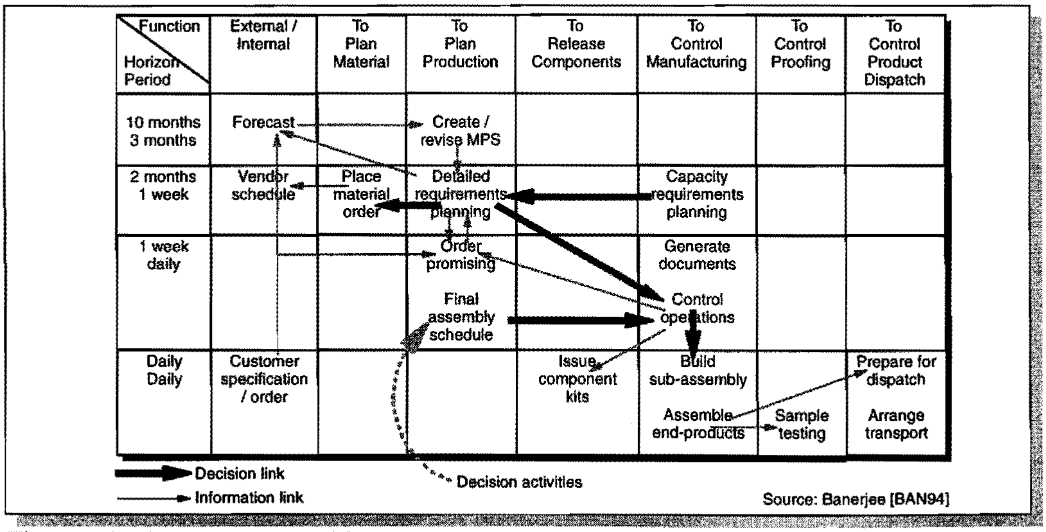


Figure 2.3 - Sample GRAI-grid

2318 Besides the graphical techniques mentioned there is a multitude of formal specification languages. One example is ExSpect (Executable Specification tool), which was developed at Eindhoven University of Technology, see among others [HEE94]. The advantages of such specification languages stems from their executability, i.e.:-

(a) the validation of the correctness of the model;
(b) the analysis of the problem area;
(c) the assessment of the consequences of the realisation of the model;
(d) the training of future users at an early stage based on a prototype.

2319 The application of specification languages in the selection of a BIS can be very useful in certain sub-areas, for instance for evaluating drastic changes such as in production planning or sales order acceptance, where the consequences with respect to the logistics performance indicators are simulated (like stock levels and delivery times). Another area lies in the framework of Business Process Re-engineering and concerns the description, simulation, and testing of fundamentally different ways of working.

2320 No supporting techniques have been found for the formulation of requirements, see **activity (c)** mentioned earlier. These requirements are normally stated in a checklist. The checklists are a limited reflection of the specifications, taking into account that the usual standard functionality can be found in each BIS. It makes no sense to formulate requirements like 'the recording of purchase orders', or 'recording stocks per warehouse', as these functions can be found in every BIS. By excluding the trivial functions, the checklist is reduced from many thousands of points to less the one hundred. One should realise, however, that these discriminating functional requirements are continually changing as developments in BIS continue to advance.

2321 Within the context of the selection of a BIS, it is essential to realise that one is not developing an information system while specifying the required functionality. The effort should on the one hand be proportional to the selection process, on the other, existing BIS already contain many ideas, 'best practices', and functions, which one does not need to re-invent. There is great risk of being unable to find a suitable BIS at all if too detailed a specification is made, having deviated from the course taken by the BIS. For as the latter point is concerned one should remember that with detailed specifications more than one road leads to Rome. Usually it is preferable to choose the road that is provided by a standard BIS.

Requirements assessment

2322 The assessment criteria that were stated, based on the functional specifications, have to be used to evaluate the potential BIS. This evaluation can be carried out in the following ways:-

(a) *by asking the suppliers concrete questions* who then provide the answers (preferably in writing). These questions are directly based on the criteria set. Naturally, the supplier is biased, but by formulating the questions as concretely as possible and by including the critical points in the contract, a certain value can be placed on these answers;
(b) *requesting information from third parties*, such as independent experts, supplier's references, and available databases with detailed information. One can generally get impartial answers from the above. However, one should realise that the information received is less accurate (of course the greater part of the expertise lies with the supplier himself) and third party information is less up to date. The latter is certainly true considering the fast developments in the area of BIS;

(c) *finding out these things by using the BIS*, i.e. by means of demonstrations, cases, and workshops. Whereas the assessment criteria are directly turned into questions in the two other ways of evaluation, here the criteria have to be built into the business case. (Assessing data structures or other technical documentation of the BIS for oneself also falls under this type of assessment).

2323 As was argued earlier, the degree of detail in such an evaluation should be proportional to the number of candidates to be assessed. Thus, the first two types of evaluation are used at the beginning of the selection process, while the last type is only useful in the final stages of the selection process.

2324 In all phases an objective consideration should always be attempted. For this purpose the assessment criteria have to be made fully explicit. The assessors have to value the BIS (and the supplier) using these objective criteria. If the criteria are not explicit, vague valuations arise that say more about the communication and commercial skills of the supplier than about the ability of the BIS to fulfil the objective requirements.

2325 In order to value the criteria set, the following categories for the BIS and its supplier's appear to work:-

(a) **satisfactory/suitable** - and therefore *fits* the criterion;
(b) **acceptable/workable** - does not entirely fit the criterion but a solution can be found that *does not involve extra costs* (for example by using a 'work-around');
(c) **adaptation necessary/actions required** - does not meet the requirement but *can be solved with controllable costs* (for example some minor modifications);
(d) **unsatisfactory/unsuitable** - does not meet the requirement and cannot be adapted, or can only be adapted with *extreme costs.*

2326 To indicate the importance of criteria they are sometimes given a scale from 1 to 10 (or an other comparable range). However, considering the scores the sensitivity of such scoring method, it appears in practice to be more workable to use 'knock-out' (if a BIS or supplier does not meet a criterion, they are immediately eliminated), 'requirement', 'wish', and 'irrelevant'. After the 'knock-out' and 'irrelevant criteria' have been evaluated, we are left with a dichotomy. As long as we are dealing with a few dozen criteria this is sufficient.

2327 If we are considering larger numbers of criteria, the following problems can occur:-

(a) *insight and objectivity disappear completely.* Research has proven that humans cannot consider and assess more than 7 ± 2 criteria at the same time [MIL56]. By creating a second hierarchical level in the criteria, in accordance with the typology in paragraph 2401 and following, for example, 50 criteria can be assessed on average. In practice this is the upper limit;
(b) *a more detailed calibration for the assessing of requirements is necessary, but quickly results in pseudo accuracy.* Thus, a total score of '920 with respect to 890' gives us the impression that the first is better. However, the subjectivity of the valuation and the assessment lead to a level of sensitivity that render such small differences meaningless.

2328 For these kinds of complex processes, models have been developed such decision support systems and expert systems. These are based on a hierarchical structure of decision criteria, in which the assessment of the criteria is top-down. Essential to the improved models is the sensitivity analysis of the score and/or assessment, see among others [MIN92]. In practice little experience has been built up with these systems.

2329 These (powerful) tools are useful in strengthening our insight into the assessment process and convincing the other parties involved. However, if the right expertise is present in the selection process, see paragraph 2249 and further, then it is not to be expected that the use of these tools will lead to a different choice.

Consequences for implementation

2400 The selection of a BIS should be based on the business concept and the strategy with regard to information technology (see the beginning of this chapter). During the evaluation of the BIS particular attention is paid to the most important functions. Therefore, there can be a discrepancy between the less important functional features of the BIS and the details of the desired working practice.

2401 There is no way of avoiding differences at the detail level. For each of these differences a choice has to be made:-

(a) *adaptation of the (desired) organisation and working practice.* There is a chance that one will have to deviate from the optimal design one has conceived. On the other hand many adaptations are of a more limited nature (otherwise one would have selected the wrong BIS);

(b) *modification of the BIS.* The disadvantage of this is that one runs the risk of becoming incompatible with newer releases of the system.

A compromise between these two options (a little bit of both) will sometimes solve this dilemma. Besides a large number of more technically-oriented activities, the BIS implementation activities are therefore often dominated with these kinds of adaptations and modifications.

2402 There is relatively little fundamental literature available about implementing standard BIS. However, in practice a number of approaches to such implementations can be distinguished. The most important types of approach are:-

(a) *the conference room pilot approach (CRP approach).* This is based on a principle dichotomy of the implementation, in which the first phase (the CRP) is aimed at a minimal effort to create a locally running system. The minimal effort is principally intended to remove all possible uncertainties for the second phase, the actual use, without having tried out the full system. In the CRP the BIS is set up, roughly configured, modifications are specified, and the procedures are roughly described. In the second phase the development of all modifications and the full customisation takes place as well as, for example, the conversion and extension of all necessary reports and documentation;

(b) *the prototyping approach.* Here, the standard BIS is set up step by step and refined until the desired result is achieved. Each step executed together with the key users consists of a limitation of the degrees of freedom of the BIS moving from rough to fine distinctions;

(c) *the full implementation approach.* The total implementation is viewed and controlled as one integrated project. In contrast to the previous approaches certain activities will commence in a far earlier stage, such as the entry or conversion of standard data;

(d) *the core system development and roll out approach.* In complex organisations one can opt to set up one central system, develop procedures, and convert all (central) standard data. After that, the implementation (roll-out) can be performed site by site on the basis of this core system. Depending on the starting point, modifications are or are not allowed to the core system and are carried through from one site to the next.

2403 In this context we will not go any further into the content of these implementation approaches. However, the implementation approach not only depends on the organisation, but also on the nature of the BIS that is selected.

2404 As we are dealing with standard software, these BIS have by definition many functional degrees of freedom, which make them applicable in as many different situations as possible. For each specific situation these degrees of freedom have to be reduced and the BIS has to be moulded into the most desirable form. The differences between the BIS and the desired form either lead to adaptations of the BIS or to adaptations of the organisation and the working practice. The decisions about where to make adaptations are based on the starting points and philosophy of the organisation and, it depends of course on the willingness of the parties involved to change.

2405 With regard to the way in which BIS can be adapted, we can distinguish a number of fundamentally different ways of customisation. In turn these types are in fact a result of the type of development tools used. In this context we have distinguished five types of customisation.

Table and parameter adjustment based on the data structure

2406 The choice of functionality within a BIS is often determined by means of variables (parameters) in the database. Thus, an item master file often contains dozens of parameters related to for example the ordering policy. The software contains the questions so that depending on the value of a parameter certain parts of the software code are executed or not. As a result the behaviour of the system is influenced. The permissible parameter values should be determined during the BIS development, so that the software code can take all possibilities into account. Considering the large number of variations in processes and the diverse interdependence of parameters, this approach leads to many additional attribute types in existing entity types and to various additional (parameter) tables.

2407 According to some experts this approach is outdated, but some form of parameter control will be inevitable in a *standard* BIS. The disadvantages include the inflexibility *after* implementation. The dynamic adaptation in an operational environment results in transition problems where fundamental alterations are made (see among others [GAR94.2]).

2408 Some important advantages are the (relatively) user friendliness of the customisation process and the independence of the supplier in this process. This leads to great flexibility during the implementation. Furthermore, new releases can in principle be implemented without much extra (customisation) effort. Finally, the technology required is simple and has matured.

Source modification based on fourth generation languages

2409 Situation specific adaptations can also be realised by changing the source code of the BIS. These can include both modifications and additional developments. Basically, it concerns traditional software development, the only difference being the far higher productivity through the use of fourth generation languages. Depending on the power of these languages, many changes can be realised with only a little effort and certainly if the core of the BIS (for example the data structure and the basic philosophy) is left untouched.

2410 One important advantage is the speed with which modifications can be realised, without an enormous diversity of functional features being present in the software beforehand. As a consequence standard BIS can be kept relatively small.

2411 One disadvantage is the installation of new releases of the BIS. Depending on the nature of the modifications, they will have to be carried out every time a new release is installed. The (technical) knowledge about the BIS and the source code generally belong to the supplier. This means that the in principle modifications will have to be made by the supplier and not by a user organisation.

Design modification based on integrated CASE tools

2412 On the assumption that a BIS is fully developed using integrated CASE tools, company specific modifications can be made at an even higher level: the design of the system. After that, the BIS is simply regenerated. This philosophy is actually identical to the development of a BIS based on (CASE) templates, as was described in chapter 1. The difference with the previous variant (on the basis of fourth generation languages) is the higher level of abstraction at which modifications are made. The changes in the design are very close to the conceptual framework of the organisation and the business processes.

2413 Important advantages are the speed with which changes can be realised in principle without (much) support from the supplier. Due to the high level at which the CASE model can be changed, it is obvious that this has to be done by (IT) specialists belonging to the user organisation itself.

2414 The disadvantages are mainly determined by the limitations of the present level of technology, and especially by the (im)maturity of the integrated CASE tools offered. These are not yet capable of generating complex BIS from a high level specification and design (see also [GAR94.2]). Linked to this is the restricted availability of high-quality templates with a multitude of 'best practice' processes, functions, and data structures. Finally, there is the disadvantage as a result of the 'big bang' generation of a complete BIS. This approach does not normally allow for changes in an operational environment.

Object enhancement based on object-oriented tools

2415 By using object-oriented technology (OOT) situation specific modifications and additional developments can be made by extending the so-called 'class libraries' with new object types. New object types are able to inherit all the relevant functionality of the object types already present, so that it is possible to focus on the changes. On the other hand, the original object types remain intact.

2416 The major advantage is that the existing object types do not need to change and therefore can be replaced later by new versions from the supplier. Because of the inheritance mechanism, the self-developed object types automatically gain this new functionality.

2417 At present a great disadvantage is the lack of OOT development environments that are mature, standardised and broadly accepted. Many BIS are converting to OOT, but are certainly not ready. It is expected that the first BIS to be completely based on OOT and fully equipped with a broad functionality will not be ready before the end of 1997. As a result there is neither much experience with the customisation philosophy described here as object enhancement.

Work-flow adjustment based on work-flow management techniques

2418 An approach that is very new to BIS is based on the sequencing of steps in the business processes based on work-flow management techniques. The functionality of the individual process steps is recorded in the database using 'stored procedures' and 'triggers'. The actual configuration of the BIS implies coupling these business rules in the desired order and on the basis of certain conditions.

2419 Major advantages are the flexibility in an operational business environment and focus on business processes instead of on a functional structure.

2420 The disadvantage again is the immaturity and the lack of broad acceptance of the technology, as well as the lack of experience with the functioning of BIS based on this in industrial enterprises. At present only one BIS is completely based on this (see figure 2.4).

Positioning and summary

2421 This classification is strongly based on the underlying technology upon which the BIS was developed. In figure 2.4 a number of important BIS have been given their relevant positions on the basis of this technology (see also [GAR94.2]).

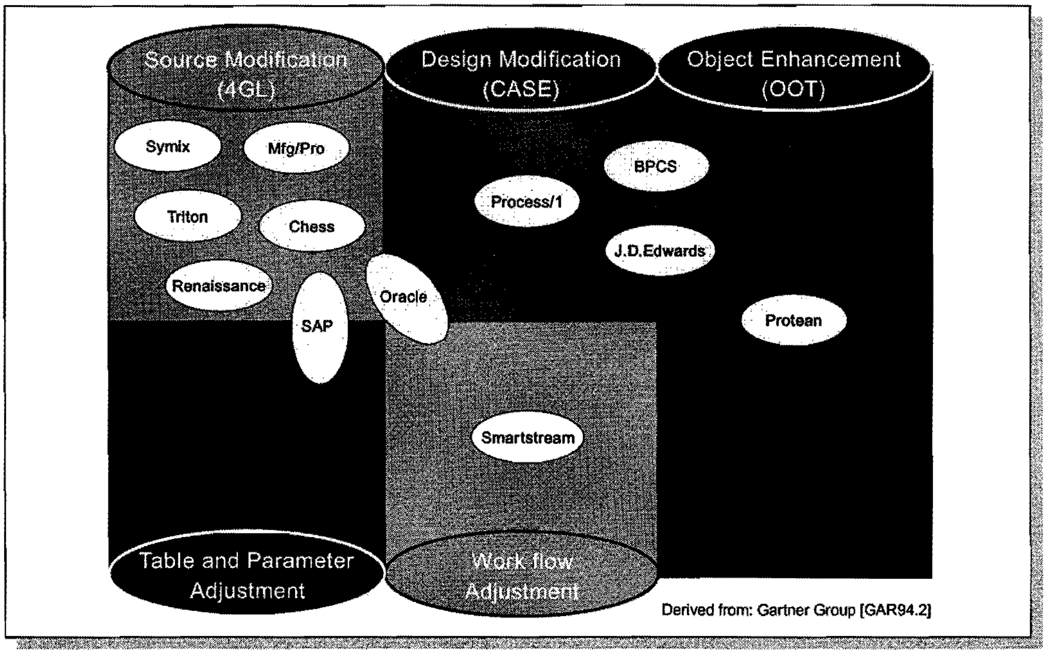


Figure 2.4 - BIS customisation techniques

2422 Additionally to figure 2.4 it should be noted however that nearly every BIS makes use of parameter control to some extent. Furthermore, many BIS that are currently based on fourth generation languages, are migrating to object-orientated technology.

2423 As discussed earlier these different forms of customisation have some characteristics that are an advantage or disadvantage in a particular form. In summary, the most important criteria for the comparison of these types of customisation are:-

(a) *the simplicity of execution of the customisation*, i.e. does the modification have to be carried out by the (external) supplier, an IT specialist from one's own company, or can it be done by a trained key user;
(b) *the ease in transferring to new releases*, without all the modifications having to be done again;
(c) *the flexibility of modifications in an operational situation;*
(d) *the maturity, standardisation, and acceptance of the underlying technology.*

	Table and Paramater Adjustment	Source Modification (4GL)	Design Modification (CASE)	Work-flow Adjustment	Object Enhancement (OOT)
Customising simplicity	key user ++	software vendor -	IT-specialist +	key user ++	IT-specialist +
New releases	+	--	-	+	+
Flexibility in an operational situation	-	+	-	++	+
Maturity of the technology	+	+	+/-	-	-

Table 2.2 - Summary of customisation characteristics

2424 The characteristics in table 2.2 are only indicative and will be somewhat influenced by the specific development tools used. The variety of CASE tools is already quite large and they certainly not comparable on all these aspects. In addition, this valuation is based on the present state of technology (late 1995), while the developments in this area are rapid.

Developments influencing the selection

2500 The development of BIS has been accelerated in recent years. It is obvious that some insight into these developments is of great importance to any company that wishes to select a BIS. For many organisations information technology is one of the tools used in competition. Certainly in those cases it is essential that the continuity of the supplier and the continual development of the chosen BIS is guaranteed.

2501 In this context, the developments, trends, and views of the various experts can be divided into four perspectives: the supplier market, the functionality, the technology, and the implementation tools (see among others [GAR94.1], [GAR94.2], and [GAR94/95]).

Developments in the BIS supplier market

2502 What strikes in the supplier market is the growing number of strategic co-operations between different parties. Especially the co-operation between BIS suppliers and implementation specialists is a clear trend. Following SAP, suppliers like Baan and SSA have made this step. The objectives that are worked on in these co-operations are:-
(a) *providing total solutions*. Traditionally, the combination of the software and hardware delivery was of great concern. Now, the provision of complete functionality for the entire supply chain, and the co-operation with parties that can take care of the whole implementation (see the examples mentioned before) are becoming normal;
(b) *each party concentrates on their own core-business,* such as the development and sales of BIS, the development of office products, the implementations, and the delivery of hardware. This leads to complex co-operatives between all these different parties. Thus, there is co-operation between SAP, Microsoft, the big six consultancy firms and various hardware suppliers;

(c) *the advanced integration of BIS with office products and other supplementary de facto standard products,* in which maximum use is made of the technological possibilities offered by the various products. Besides the Microsoft products, for example the Aris-Toolset has been adopted by various BIS suppliers.

2503 The co-operation between parties that offer additional BIS modules in many cases will lead to the full integration of these companies, through a take-over for example. Examples of such 'take-overs' are JBA with Ratioplan, Oracle with Datalogix, and Baan with Berclain.

2504 An increasingly important consideration is the financial power and continuity of the BIS suppliers. Due to the very rapid developments, suppliers are being forced to make enormous investments in technology, functionality, *and* implementation tools. These investments have to be financed by the sales and maintenance of BIS. If sales fall because of the enormous competition, and the existing installed base moves to more modern systems, a considerable number of suppliers will no longer be capable of supporting these investments. As a consequence the experts expect a shake-out of the BIS supplier market.

2505 On the other hand, very advanced development tools coming from the low-end of the (PC and Windows) market. The expectation is that new players will enter this BIS market, who are not impeded by the existing installed client base with all their requirements in the area of upgrading and support. These new BIS will be characterised by specialist functions at low prices and will make use of all the most modern techniques available (especially OLE and OOT). On top of this, these new BIS will be integrated with standard office applications from Microsoft, Lotus and others.

2506 The expectation is that industrial manufacturing enterprises will be able to choose between a large variety of BIS in the future. Only for extremely complex engineer-to-order and make-to-order situations the supply will remain more restricted for the time being. The total market for these BIS is expected to double in the coming five years (see also [GAR94/95]).

Developments in the functionality of BIS

2507 In chapter 1 (see paragraph 1111) a number of functional features which characterise ERP-systems were already mentioned. This does not mean that the present generation of BIS already covers all this functions, but that these are the spearheads in the present developments. The features mentioned included full functional integration, decentralised multi-site structures, flow-based manufacturing, support of hybrid manufacturing and distribution environments, planning based on financial units and performance indicators on top of a main production plan, logistics chain integration with customers and the specific functions for customer-order controlled manufacturing such as product configurators.

2508 Another development concerns specific functionality for semi-process industries. Until recently, the suppliers for this part of the market were mainly the traditional MRP systems with some functional adaptations, and a few specialised systems that use the production model as a kernel (comparable with a recipe). Particularly, we referred to Prism by Marcam, Gemms by Datalogix, and Renaissance by Ross.

2509 In the mean time, nearly all the important players have tackled this specialised functionality. SAP has developed a very extensive module called PP-PI, Baan has purchased the package Probe and has developed the new BIS Triton-Process, and J.D. Edwards has also developed a specific module for this segment.

2510 Another trend is the development of branch specific configurations (also known as pre-customised versions or templates). Thus, templates have been developed for Triton, for example, for the automotive and engineer-to-order branches, while SAP has business templates and configurations (some of which are still in development) for oil & gas, retail, banking, government, and trade.

Developments in the technology of BIS

2511 The technology behind BIS is of growing importance. This is due to the broad range of functionality that is already available. Apparently, the (assessment) criteria are shifting from functionality to technology. Basically suppliers have to choose from two different approaches for introducing new technologies:-

(a) *a modular migration from the existing (broad) functionality.* At first most functionality will still be based on the traditional technology. Module by module a transfer is made to the new technologies (examples of this are System'21 and BPCS);

(b) *the development of a completely new BIS based on new technology.* The new BIS is developed and sold along side the existing BIS, until all functions are fully available in the new BIS (examples are Protean and Smartstream). The consequence of such new BIS is that it will be not support the full range of functions for quite some time.

> *Due to rapid (technological) developments, qualities of Business Information Systems, proven by implementations, are by definition outdated.*

Thesis 2.1 - The rapid technological development of Business Information Systems

2512 In many organisations philosophies like 'continuous improvement' and 'business process re-engineering' are on the increase and as a result the required functionality is changing. An implemented BIS should be able to adapt to these continually changing requirements. However, the technology at present is not capable of dealing with the requirements on flexibility in an operational environment (see also figure 2.4). During implementation a lot of flexibility is available, like the use of parameter and table control, but once the BIS is operational fundamental changes in the structure lead to complex (or impossible) conversion problems and version control. Only in the long term the development of CASE tools in combination with OOT might offer a possible solution to this problem (see also [GAR94/95]).

2513 Finally, a number of experts expect that the in-house development of BIS may again become an alternative (see among others [GAR94/95]). The development tools being made available, are making it feasible to develop such extensive and complex systems. The effort, expressed in time and money, is becoming comparable to the implementation effort of a standard BIS. One marginal note has to be made though, that the right knowledge (for example 'best practice' experience) has to be available to develop a functionally correct BIS, while this expertise is already included in standard BIS. Next to that, there is a number of on-going developments that will help to simplify and enhance the implementation of BIS.

Developments on the implementation of BIS

2514 The implementation of the present generation of BIS is a sizeable and complex activity. In combination with the increasing demand for BIS, like SAP and Triton, this has resulted in a shortage of experienced implementation consultants for these extensive BIS. This shortage of expertise has grown at present into a global problem. This has led to relatively high fees for these experienced consultants, but also to a large number of so-called 'experts' who have suddenly appeared, causing delays in implementations.

2515 Partially because of this, there is growing interest in tools that will simplify the implementation of these BIS. Examples are the Aris-Toolset, which contains the complete SAP model. The changes of this model are incorporated in the on-line Implementation Management Guide (IMG) with which customisation of the system is supported step by step. Triton also has a number of modules that are grouped together under the name Triton-Organiser, with which the installation of Triton can be controlled and executed via among other things templates.

2516 Due to the competition, smaller organisations also require the broad functionality offered by the BIS mentioned here. As a result, we are now seeing the development of 'Lite' versions, branch models, and templates. These are cheaper systems that are easier to implement and that are also suitable for the lower-end of the market.

3 Analyses and problem definition

3001 The success of an implemented Business Information System (BIS) is mainly determined by two important factors, i.e. the degree to which the system suits the desired business situation (the fit) and the quality of the standard BIS implementation in combination with this business situation. By quality, a combination of aspects is implied including project management, change management, involvement, and commitment. An optimal implementation project is capable of correcting many imperfections from previous phases as well as restrictions in the functionality offered. Nevertheless, a better fit does mean a considerably smaller work load during the implementation requiring less software modifications and procedural changes, so that one gains a better starting position for additional developments and new (software) releases.

3002 In this chapter, from paragraph 3100 and further, an analysis will be made of the customary ways to assess the BIS functionality. From this analysis it appears that the number of risks and limitations involved in assessment are a due to lack of insight, or to being incapable of gaining this insight during the evaluation process. This is a direct result of the 'black box' approach on which the existing assessment methods are based.

3003 From paragraph 3200 onwards an elaboration of the differences in approach between a 'black box' approach and an approach that is based on knowledge of the internal (system) structure. The data model being most important component in this internal structure.

3004 Next, paragraph 3300 and following go into experiences with and research into data structures in relation with the assessment of BIS. Here, prior to the formulation of the research problem attention is paid to the potential risks and opportunities offered by data structures as an assessment mechanism.

3005 Finally, from paragraph 3400 onwards the research objective and approach are formulated.

Analysis of current methods for assessing functional requirements

3100 In the previous chapter the current methods for assessing functional characteristics of BIS were described extensively. Two approaches can be distinguished, namely:-

(a) trying out (testing) the BIS by means of a company case;

(b) evaluating the answers from suppliers or other experts to a (preferably written) questionnaire.

Both approaches are analysed consecutively.

Analysis of the testing of systems

3101 While testing BIS the quality of the test results very much depend on the quality of the test case. In simple and locally used software packages, a general demonstration may be satisfactory. Due to the enormous complexity and broad functionality of the present integrated BIS, an extensive, company specific case is necessary for such a test.

3102 This way of assessing or testing a BIS is comparable to the testing of software in systems development. In the first case, assessing of a BIS, the test is aimed at whether the system functions properly compared to the desired requirements. In the second case, testing of software in systems development, one is interested in whether the system functions correctly with regard to the design specifications. However, there is also a principle difference, i.e. in the second case it is also important to determine whether any design or programming errors are present, while a standard BIS can be expected to be error free (though in practice this may not always be the case).

3103 Continuing the analogy of testing in systems development, one is confronted by a very large number, nearly infinite number, of test routes during the testing of full company situations. These test routes describe all desired functions, business processes within a company and a large number of situations with their peculiarities and exceptions. This means that a test case has to be restricted to the main business processes and their most important characteristics.

3104 It is assumed then that 'the rest' will work, as is being dealt with a standard system that functions in many companies. Nevertheless, in practice one is regularly disappointed. Many software packages often have quite a narrow installed base for one specific version and some functions are not used as widely as one may expect.

3105 Problems occur regularly at the interface between modules, which cannot be easily traced during testing. Each software package has its own module structure. Modules sometimes are bought from third parties and therefore have a different set-up and terminology. Instead of the complete integration of modules one regularly sees the necessity for double input, redundancy of data between modules, validations that are not made, and batch interfaces. It appears particularly necessary to test all possible routes as the module boundaries.

3106 It should be obvious that a pragmatic selection method leaves too little time to carry out a complete and intensive test of all the candidate BIS. Certainly where the case demonstration takes up the usual number of days per software package, it seems that users need the demonstration time to become familiar with the terminology, working practice, and the user interface. This is why graphical user interfaces are such a commercial success. Besides the demonstrated functionality, it seems that more important aspects are:-

(a) the commercial skills of the demonstrator;
(b) the outward appearance of the system;
(c) the extent to which the system links up with the existing frame of reference and the information system to be replaced.

3107 Some clever demonstrators are capable of giving a seemingly flawless demonstration of functionality that is not present for example, by operating different versions of the system, prepared with the same case data, in separate windows and then 'secretively' changing windows during the demonstration.

3108 For reasonably simple and orderly software packages, the solution may be to test the preferred system intensively with an extensive company case for a short period of time. This means that before a contract is signed, the test is carried out full time for several weeks. Such a test provides insight into the suitability and gives an overview of the possible modifications and customisation that is necessary. A test carried out in this way is also known as a Conference Room Pilot (CRP), though this term has also gained several other meanings.

3109 In more complex systems, such as all ERP-systems, a short and orderly CRP lasting a few weeks is no longer possible. The present BIS consist of thousands of program functions with a multitude of options. By means of a great many parameters (sometimes ranging into the thousands) the operation of the functions is controlled. Setting these parameters in itself sometimes takes months of specialised work. This seems to be an inherent result of the developer's endeavour to provide the BIS with the maximum amount of functionality, in order to cover as large a market as possible.

3110 Naturally, if one were to believe the supplier, the software provided is perfectly suitable, completely integrated, flawless, etc. Unfortunately, practice often proves different. Software is never perfect and never will be. However, the application of modern object-oriented development tools makes the testing of individual objects possible and, in contrast with the traditional modular structure, the integration aspect becomes more controllable. One should realise, however, that the number of objects in a complete BIS will be very large.

Analysis of questionnaires

3111 Another approach in the assessment of the functionality is based on questionnaires, which are answered either by the supplier or by some other package expert. Whether or not the questionnaire consists of knock-out criteria or is simply a list of detailed points, one always has to take a number of characteristics and risks into account with their use.

3112 The unambiguous written formulation of questions and requirements, and therefore the interpretation, is tricky. On the one hand being able to clearly define the requirements, and on the other hand understand someone's text, depends on one's writing skills, the use of terminology, and the frame of reference of both parties. Written communication about such complex matters can quickly lead to many misunderstandings.

3113 One should also realise that if a positive answer is given to a number of questions, it does not mean to say that the combination also works. Both in the use of 'work-arounds'[7] and in parameter-controlled software, the use or activation of one option may mean that another is no longer available. A simple example is the control of commercial items: an item is either stock-controlled or customer order-controlled, the combination of both, for example dependent on the size of the order is more difficult and is therefore not automatically supported.

3114 It is essential when writing a good questionnaire to restrict it to the most important points. This reduces the amount of work so that the general quality of the answers increases. In addition, one creates an overview that considerably simplifies the evaluation. However, scrapping unimportant questions proves in practice to be quite a difficult task. Each person involved tries to make as good and as extensive a contribution as possible, so that questionnaires have a natural tendency to expand. It is difficult to distinguish the important from the unimportant questions. There are two types of questions:-

(a) when formulating functional requirements there is the *category 'nice to have'*. This kind of functionality does not necessarily need to be present in the BIS, the requirement should also hold for the foreseeable future. Though everybody tries to make this distinction at first, it turns out that people tend to add more and more wishes to the requirements list the further they get into the selection process. For that matter important requirements for the organisation do not always have to be important for the evaluation of software packages.

---

7 A 'work-around' is an unintended (or figurative) use of possibilities within the software. Work-arounds are used in almost every implementation of BIS and are normally quite acceptable.

Sometimes simple 'work-arounds' are possible, providing an equivalent (or even better) solution. Sometimes a small modification of the software is all that is necessary to realise a requirement. To make this distinction, one needs knowledge of both software development and the structure and philosophy of the software package in question;

(b) from the perspective of the BIS there is a large number of *functions present in every software package*. These are the non-discriminative requirements. Thus, each logistics system supports a purchase function, and purchase orders can contain multiple items to be purchased and the goods receipt function is always supported. The more software packages develop, the more the discriminating requirements shift. What used to be a requirement in many questionnaires (for example contract agreements with customers and suppliers) is nowadays available in any reasonable software package. The risk remains, however, that there is an unexpected gap in the system. A typical area at risk is the linkage and integration between modules, for example an acquired marketing information system, or the link with a CAD system.

3115 An entirely different distinction can be made with '*open questions*' versus '*closed questions*'. Open questions give the supplier or package expert the opportunity to give an extensive reply, by listing for example alternative options or solutions that the software package has, thus meeting the functionality requirements. However, the result very much depends on the supplier's effort and his creativity in indicating alternatives. An important disadvantage is the risk of receiving extensive answers that add no further information, are difficult to asses, and often consist of general sales pitch.

3116 The closed questions have to be answered with 'Yes' or 'No', or with a qualified 'Yes' where multiple choice questions have been poses. These questions have the important advantage that the assessment and comparison of answers is very much easier. Directly opposed to this, is the fact that this assessment gives less insight into the specific philosophy, terminology, and other characteristics of the software package.

3117 Perhaps the most important type of question wording is the distinction made between questions concerning the extent to which a (specific) solution is present in a system, versus questions that ask whether a certain problem can be solved using a system. A number of remarks can be made about both types of questions.

Questions that formulate a solution

3118 Questions that inquire as to the presence of certain solutions (or implementation in the software) can be expected to have a clear and concrete phrasing. This will normally hold for the given answers. However, the problem is that no inquiries are made as to alternative solutions, particularly because some will be overlooked or are not even known. Depending on the intensity of use, alternative and often simpler solutions could be far better, while a more frequent occurrence of the problem may make more extensive solutions preferable.

3119 An example: for the purposes of the purchase function a question could be asked about an option to handle fixed delivery schedules. In an advanced just-in-time (JIT) environment with excellent supplier relations, this is an important part of the JIT puzzle. If, however, we are dealing with just a limited number of call-off orders one could also consider using several purchase order lines for the same item with different delivery dates. The question could then be formulated as follows: Can an item occur in multiple purchase order lines within one purchase order, and can the delivery dates differ?

3120 It occurs regularly that when writing up the questions other, possibly better, solutions have not been considered. A short and honest answer to such a direct question may eliminate a supplier from the selection, while a good alternative solution may exist in the software package in question. The assessment of the quality of solutions is extremely difficult and requires much knowledge and experience in the area of systems development, as well as in the application of systems. More or less elegant 'work-arounds' can sometimes offer a good alternative too. Their assessment requires considerable knowledge about the software package in question.

Questions that formulate the definition of a problem

3121 Questions that formulate the definition of a problem and request for any solution to this problem, are often propagated by consultants. However, a positive answer („Yes, the software package provides a solution to this problem") has a limited value. Besides being it extremely difficult to formulate a compact question around a complex problem, the quality of the proposed solution remains essential. Some of the solutions offered may be very awkward (though there is a 'solution' or 'work-around'), while other solutions may not operate under certain conditions. What one would actually like to see is a brief discussion of all the alternative solutions and an indication of all the consequences of these alternatives. Obviously, a problem of some complexity will lead to sizeable answers.

3122 The 'work-arounds' mentioned earlier are quite important here. Though suppliers are good at thinking up creative solutions, it requires a lot of inventiveness, and knowledge of and insight into the system. Deviating from the normal routes in the system can lead to unpleasant surprises in totally different areas (functions).

3123 A real example: The transport time from the supplier to the receiving warehouse is not taken into account within the system. However, the goods are delivered ex-works however, which means the supplier has to be assessed on this delivery date. Before the goods are available at ones own warehouse, some time will have elapsed and the receipt date differs from the delivery date. This is certainly the case if the goods are from far afield. An example of a possible 'work-around' is to use the 'goods receipt processing time' as the transport time, so that the goods do not enter the stock on the planned receipt date but only after the 'goods receipt processing time'. Another 'work-around' is the introduction of a separate transport warehouse where the goods remain during transport. Naturally, 'work-arounds' very much depend on the system that is used and it is not possible to go into all the consequences here.

Other risk areas in the use of questionnaires

3124 A clear restriction in the assessment of the functionality offered, and certainly on the basis of questionnaires, lies in the fact that little attention is paid to the ***undesirable functionality***. An extreme example is that of a small company that makes a limited functional questionnaire and has it answered by a supplier with the most extensive software package. The system will be capable of handling everything that is required, but it is capable of so much more that an enormous ballast will be purchased without this becoming clear from the answers to the questionnaire.

3125 More realistic situations where surplus functionality is purchased often occur in practice. The disadvantages, risks, and (hidden) costs can be related to:-

(a) unnecessary extra strain on the hardware;
(b) extra effort in the installation and implementation of the software package;
(c) additional training because of the system complexity;

(d) unnecessary handling when the system concerning unused data fields, screens, process steps, and functions;
(e) tendency to cultivate sloppy working practices in daily use.

3126 A specific problem with long lists of questions is *the lack of structure and accessibility.* A list has only a one dimensional structure and when the number of questions starts to range in the hundreds, the problems mentioned will arise. For example, should all the questions concerning tools be combined or should everything to do with planning be combined instead? Or should standard questionnaires be structured according to the production typology? But in both semi-process and engineer-to-order industries network structures of operations can be found!

3127 *One should not deduce from the above that the use of a questionnaire is wrong, but one should realise that making a good questionnaire is no easy task.* The problem is that people tend to want to solve the restrictions of a questionnaire by adding more and more questions. Though this may seem to be a natural tendency, this spiral results in unmanageably long lists of questions. In practice it is wiser to overcome these restrictions by means of a concise questionnaire combined with a good test case.

3128 Directly related to this is the fact that one is not capable of making proper judgement based on long lists of criteria and arguments. Though one intuitively tries to rationalise the selection process by adding criteria, the result is that the final judgement becomes less rational. It appears from research that one can only deal with 7 ± 2 criteria at the same time [MIL56]. At best, one will arrive at the same short list of BIS with such a detailed questionnaire as with a rough selection carried out on the basis of knock-out criteria. However, there is a bigger chance that a more emotional choice is made.

Conclusion from the analysis

3129 From the analysis of both approaches described earlier, active testing and passive questionnaires, the following main problem remains: the lack of insight into the operation of BIS. When testing a BIS with a company case, the selection process is too short to gain this insight. For complex BIS such insight will only be built up after far more extensive experience with the system and usually only after the implementation. For simpler packages an isolated 'Conference Room Pilot' lasting one or two weeks as the final part of the selection may provide enough insight into the BIS. A pure study of the written answers to questionnaires certainly do not give one any of the required insight. The philosophy behind the package is very much concealed behind a multitude of 'Yes's' and 'No's'.

3130 In order to make the right choice it is important to understand how the BIS is really set up, what the, for example logistics, philosophy is behind the system and what the designer had in mind during the development of the system. This insight makes it possible to effectively avoid the risks and limitations mentioned earlier. Specifically, having this insight means one is able to:-
(a) *ask the right questions.* Understanding the philosophy and structure of the package makes it possible to point out weak spots;
(b) *predict the direction answers will go in.* Without knowing exactly what the answer will be, an insight into the structure of the BIS will help separate the wheat from the chaff (this ability is usually hidden within the term 'experience'). People are able to judge the quality of the answers;
(c) *assess the consequences of alternatives, solutions, and 'work-arounds'.*

3131 The conclusion is that both of the approaches dealt with provide insufficient insight during the selection process. The cause lies in their 'black box' characteristics, i.e. the assessment of complex systems from the outside. One could compare this to situations in our daily lives, like taking a new car for a test drive (this analogy is worked out more detail further on). If one wants to gain insight a solution may lie in the assessment of the internal structure of BIS.

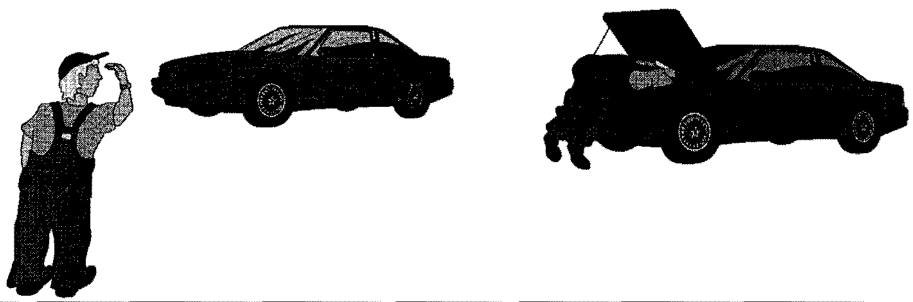


Figure 3.1 - 'Black box' approach versus assessment of the internal structure

3132 As a result of gaining insight the assessor (eventually) has more expertise with regard to BIS than by using questionnaires (of course the question is then is what exactly caused the improved quality of the assessment: the method of assessment or the knowledge built up by the person that developed and uses the method?).

'Black box' approach versus assessment of the internal structure

3200 In the 'general systems theory' a general research approach has been developed. This approach is so general that it is applied in areas such behavioural sciences, psychology, and biology, but also in cybernetics and information theory. On the basis of this theory, systems can be defined as *collections of objects with a mutual relationship that distinguishes them as a group from the environment*. For a complete definition and description of all the related terms the reader is referred to [KRA79].

3201 The 'black box' [8]model is a part of the 'systems theory'. It describes a method of systems research in which there is no interest in or knowledge of the structure of the system under investigation. *'Black boxes' can be defined as systems of which the content is unknown or which is uninteresting to the observer and of which the relationship with the environment runs according to fixed patterns* (see also [KRA79]). The behaviour of a 'black box' is studied by relating the output to the input of the system.

3202 In the investigation of BIS, a large number of questions and questionnaires, but also the testing of software (see paragraphs 3101 and following) fall under the heading of the 'black box' approach. Typical examples of 'black box' questions are: Is certain information (output) available? What happens (output) in a certain situation (input)? Is a certain function (output) possible? These are process or behaviour-oriented questions and in principle this type of question is equivalent to the kinds of questions that help formulate a problem definition problem (see paragraph 3121).

8 The term 'black box' has arisen from the electro technics, where in the past connections were moulded in black bakelite. To repair these black box 'systems' a technician used a black box approach to find the error and to replace the box concerned.

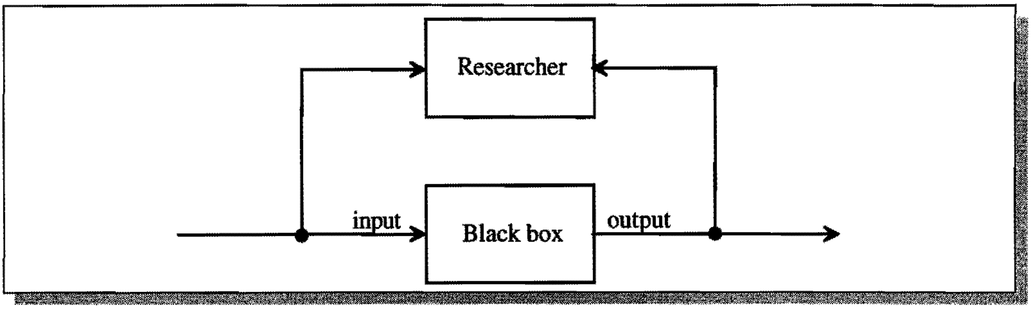


Figure 3.2 - 'Black box' assessment

3203 It is characteristic of this approach that the results from test cases (input-output relation) are used to extrapolate this to all comparable situations. It is not possible to verify an infinite number of situations. For each type of situation (in as far as one wishes to assess it) a conclusion is drawn with regard to how all other possible cases may function on the basis of a test case.

3204 The 'black box' approach *must be used* if one is incapable of assessing the content of the system, or one is unfamiliar with it. However, computerised information systems have a structured design, are developed with mechanised tools, and undergo systematic maintenance and renewal. Software systems (and thus BIS) are essentially composed of structures defined by basic elements (the formal language) of the system software. The components (of the computerised part) of a BIS are:-

(a) *program structure:* the dynamic behaviour structure of software systems. This structure is completely described in a code that is eventually interpreted by the system software. So, depending on the system software the structure is described in a diversity of languages, such as procedural languages (Pascal), algebraic languages (APL), object-oriented languages (Delphi), and schematic languages (Data Flow Diagrams);

(b) *data structure:* the static behaviour structure of software systems. This structure is described in a dictionary of attribute types, entity types, and relationship types. At a higher level of abstraction, data structures can also be represented schematically. For this purpose languages are available, such as like ERM (Entity Relationship Model by Chen [CHE76]), NIAM (Information Analysis Method by Nijssen [NIJS89]), and IDEF1X (ICAM Definition Language by the U.S. Air Force [BRU92]).

3205 The system software thus offers the basic elements, the tools with which these static and dynamic system components are developed, such as the operating system, the database management system, the programming languages, the case tools, and the tools for defining the user interface. These tools only determine the behaviour of the software system to a limited extent. They predominantly influence aspects like user friendliness and response times and therefore do not influence the functional behaviour of the system. Though theoretically the functional options could be restricted by the choice of system software: about twenty years ago complex arithmetic functions in a programming language as RPG were practically impossible.

3206 The types of questions regarding chosen solutions in BIS (see paragraph 3118) are directly concerned with the internal structure. Such questions are related to the data structure, the program structure, or both. From the analysis (see paragraph 3129 and following), it appears that posing questions about the internal structure (i.e. whether or not a particular solution has been implemented or not) is not sufficient in itself to acquire the necessary insight. The problem areas mentioned earlier with regard to questions about the internal structure are:-

(a) usually, various alternative solutions are possible and one would have to inquire about all of them;
(b) it is possible that the one asking the questions overlooks certain solutions;
(c) the answer often rises new questions, which leads to an interactive tree structure of questions and answers.

3207 *A proper approach seems to lie in the formulation of the question in the form of a problem definition: the desired output resulting from particular input ('black box'). The answer then should be given in the form of structures, being the internal structures of the (alternative) solutions that the system offers.* Based on this answer, the questioner should be capable of deducing the quality of the solutions offered in relation to the question. Consequently, special skills are required of the questioner. In its extreme, the answer to the question whether a BIS is suitable for a particular company situation (being 'the question') lies in the program code and the data structure of the system (being 'the answer'). Though this may be true, it is not directly applicable. Both 'the question' and 'the answer' have to be broken down into manageable parts. To elaborate on 'the answer', one endeavours to clarify the internal structure of BIS.

3208 As far as the dynamic structure of the software system is concerned, a lot of research has gone into mathematical methods to demonstrate the formal correctness of this behaviour [DIJK84]. In theory such formal methods are perfectly suitable for the assessment of BIS. However, a fundamental problem is the lack of an exact formal definition of the problem. Besides, the size and complexity of BIS make it impossible to mathematically prove its correctness. There are no successful applications of this method on a large scale during the development of BIS, nor that it could provide some assistance in assessing BIS.

3209 The use of formal specification languages (like VDM [JON86]) is to far removed from the assessment of BIS due to their level of detail. However, a number of these languages do offer simulation capability, so that the suitability can be tested using practical examples. One important condition for application in BIS assessment is that the BIS are described with the aid of a formal specification language. However, to the best of our knowledge none of the available BIS have been described this way.

3210 The static behaviour structure of software systems, defined in the data structure has far more to offer. Much research has been carried out in the area of data modelling, but little that is relevant for BIS assessment (see paragraph 3300 and following). One important element here is the *structure* of the data and not just the study of the (enormous) collection of data elements (see figure 3.3). See appendix B for the notation conventions used.

3211 From this analysis the following summary can be made:-
(a) the function testing of a BIS is typically 'black box' approach, as defined in 'systems theory';
(b) the results of such a test can be extrapolated to comparable situations, so therefore one can assume that in those situations the BIS will react in a similar manner;
(c) questions regarding the functionality of a BIS often define a problem situation, which makes them equivalent to the 'black box' approach;
(d) however, these questions can ask for specific solutions and therefore may have more to do with the internal structure of the system;
(e) program structures and data structures describe the internal structure of software systems and are an answer to both types of questions.

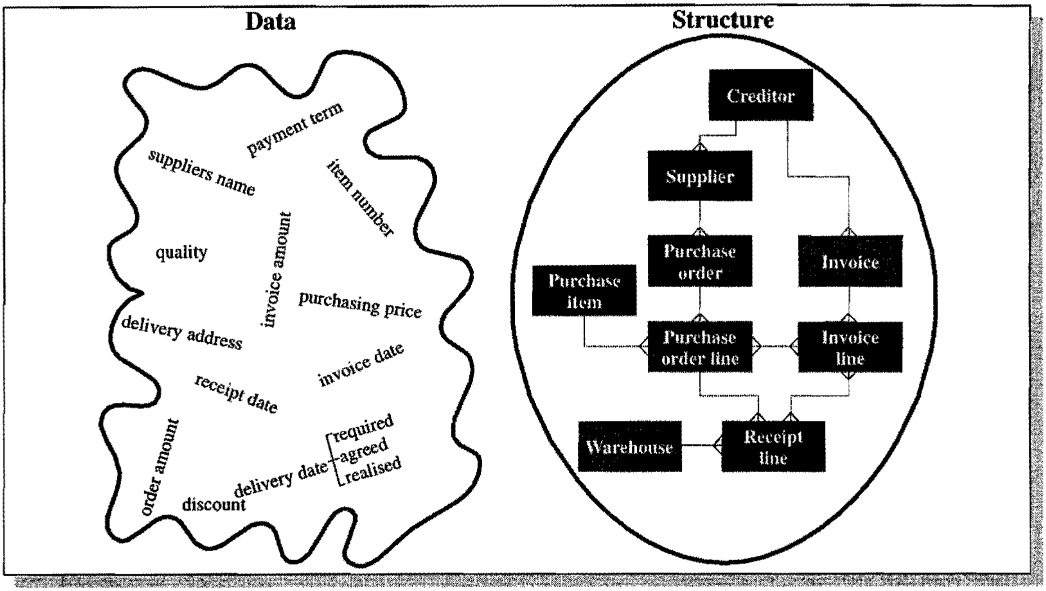


Figure 3.3 - Structured data in Business Information Systems

3212 Data structures can therefore offer an answer to questions and requirements concerning the functionality of BIS. This assessment method appears to be unexplored territory.

3213 By studying the data structure of BIS one would like to be able to make statements about the suitability of the BIS on the following business levels:-
(a) the physical features of the market, products, goods flow, and physical processes;
(b) the working practices of the organisation, the business processes;
(c) the control structure of physical and business processes.
These aspects should be recognisable to some extent in the internal structure of the BIS.

Assessing the data structure opens the 'black box'

3300 Broad experience is available in data structure modelling, see for example the multitude of BIS. These experiences may provide support when using data structures as an assessment instrument. Here, an evaluation of experiences is given in the usage of data structures particular regarding this research in. After that, a number of potential weaknesses or risks are discussed, which are involved with only the observation of data structures.

Experiences with and research into data structures

3301 Data models have been developed for just about every business situation, mainly for the in-house development of information systems and of course for the development of standardised BIS. In addition, various data structures have been designed for educational purposes or as a reference for a particular type of business or industry. The objective of these data structures always is to make a specific model for a particular situation or a generally applicable model for a type of situation: a universally applicable data model within the scope chosen. One example of a generally applicable data model is described in 'A production management information model for discrete manufacturing' [RAY95]. Here, a data model for the non-continuous industry is described.

3302 Regarding the use of data structures in the assessment of BIS, where the whole point is to interpret existing data structures, experience are scarce. Although the approach of this particular use of data structures is stimulated by experts within information science, it is hardly ever applied. The few practical experiences to date have had little success because:-

(a) the data models set up were too general and therefore express very little;
(b) the data structures of standard BIS are requested for at an early stage in the selection process, so that many candidates provide a sizeable and often poorly legible amount of documentation, which is difficult to assess efficiently;
(c) the data structure is formulated for a very specific business situation, the structure and the terminology used are special, so that comparisons with the data structures of other BIS are almost impossible;
(d) the data structures of standard BIS are not present, not available, or not accessible;
(e) any frame of reference for the comparison of such complex data structures is lacking.

3303 In 'Buying software that fits' [POR92] a short overview is given of why data structures are important in the assessment of production control software. Besides the fact that the scope of the article is rather limited, unfortunately no reference is made to (successful) applications. No details are given on a solution to the problems listed above. In a Dutch article [STA95] the use of data structures is also recommended, but it also lacks any references to practical applications. Considering the timing and content this article it probably follows from the publicity surrounding this research (see among others [HEI94.1], [HEI94.2], [HEI94.3]).

3304 Finally, a comparison was made of the available software packages for Shop Floor Control on the Dutch market in 1995 [ITC95]. It describes the data structures of most of the packages assessed. The addition of the data structures in this evaluation might also be related to this research. Though the published data structures are a useful extension to the evaluation for some experts, in the survey no actual application of the data structures was made. No assistance is provided for a comparison or other way of assessing these data structures.

3305 A few related studies and developments are taking place. Relevant in this area is the development of the Aris-Toolset by IDS-Scheer [SCH94]. With the help of this toolset, data, function and organisation structures can be modelled in relation to each other. Besides this application, which is, in principle, is possible with any case tool, the toolset offers functions to compare different models with each other. As a foundation, IDS has developed some reference models for different industries. The SAP-R/3 and System'21 models were also set up using the Aris-Toolset. The SAP-model, available as SAP-Analyser contains only a meagre abstract of the SAP data structure, which is insufficient for the assessment of this BIS.

3306 As a development extension to the Aris-Toolset and reference models as a foundation, A. Hars [HAR94] investigated the comparison of data structures. The starting point in his research is a collection of reference data structures that have been developed, for instance with the aid of the Aris-Toolset. To compare different data structures, he developed a generalised technique on which existing techniques for modelling data structures (like ERM, NIAM, and others) can be projected.

3307 The research of Hars is dealing consecutively with the selection of (partial) data structures from a number of reference models, the adaptation of these data structures for a specific company situation and the integration of the (partial) structures into a complete data model. Though the selection of (partial) structures is related to our research, Hars restricts himself to the presentation, condensing and (syntactical) translation of data structures.

3308 Hars's research provides a good survey of existing knowledge in the area of data modelling and describes a series of techniques to provide extra insight into, and to combine and extend, existing data structures. However, the work is limited to the syntactical aspects, while our objective is specifically searching for the knowledge and expressiveness in data structures that is hidden in the semantics.

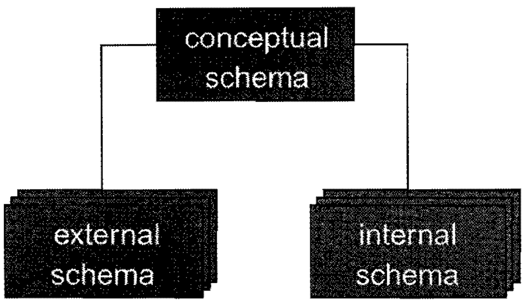
3309 The Aris-Toolset is an interesting tool, but it offers too few options as yet to compare essentially different data structures, for example the company situation and the BIS to be tested, each with their own specific terminology and definitions.

3310 Finally, a number of investigations were carried out on standard (reference) models for specific industrial sectors and production control situations. One example is the ESPRIT project AMICE, where work was done on an Open Systems Architecture for Computer Integrated Manufacturing (CIMOSA) [ESP91]. Within this context reference models have been developed that are far broader than the data structures alone. Such reference models combine much of the knowledge of characteristics relevant to this research in the area of logistics and production control.

Potential risks of data structures in the assessment of BIS

3311 The data structures found in BIS are the final result of considerable software development. These implemented structures are in ANSI terms the internal model [ANS75] (see figure 3.4). The conceptual model was probably the foundation for the development and realisation is generally no longer available.

3312 Developers of BIS probably design a conceptual model initially, but after realisation only the internal model is left. Maintenance is carried out on this model and based on this, the BIS develops to new releases. Though the conceptual model may seem interesting to find out about the possible functionality on the one hand, the transition to the internal model brings a number of choices and changes with it that influence the options that are offered. In any case, one has to take into account the various changes made to the conceptual model to accommodate speed, limitations of the DBMS, and simplification.



The external database schema's describe the views in which the data is presented to the different 'users'. By the way, a 'user' might also be a software module. The internal database schema shows the stored data format: the implementation of the data structure. The conceptual level describes the logical data structure which is in principle a direct representation of the reality.

Figure 3.4 - ANSI/SPARC - three-schema architecture

3313 A typical implementation, especially in the older BIS, concerns the combination of several entity types into one physical table/file. Using so-called 'record types' and indicators, unravelling the data takes place in the program structure. Studying the data structure without the program structure can lead to distortions. In more modern BIS such implementation choices can also lead to these kind of distortions.

3314 Another risk in studying data structures is that not all functions suggested by the data structure actually have to be realised by the program structures. This risk hold particularly for attribute types. The description 'for future use' can sometimes be found in the documentation on data structures. On the other hand, a proper design of the data structure implies that extra functionality can be added relatively easily. How one deals with this risk should become clear during the course of this research.

3315 Furthermore, not all functionality is embedded in the data structures. An obvious example concerns the complex algorithms that are controlled by a minimum number of parameters. In the development of BIS different choices can often be made for the implementation of functionality. These choices are concerned with realisation in the program structures and realisation in the data structures. Figure 3.5 gives a number of aspects that play a role in this consideration.

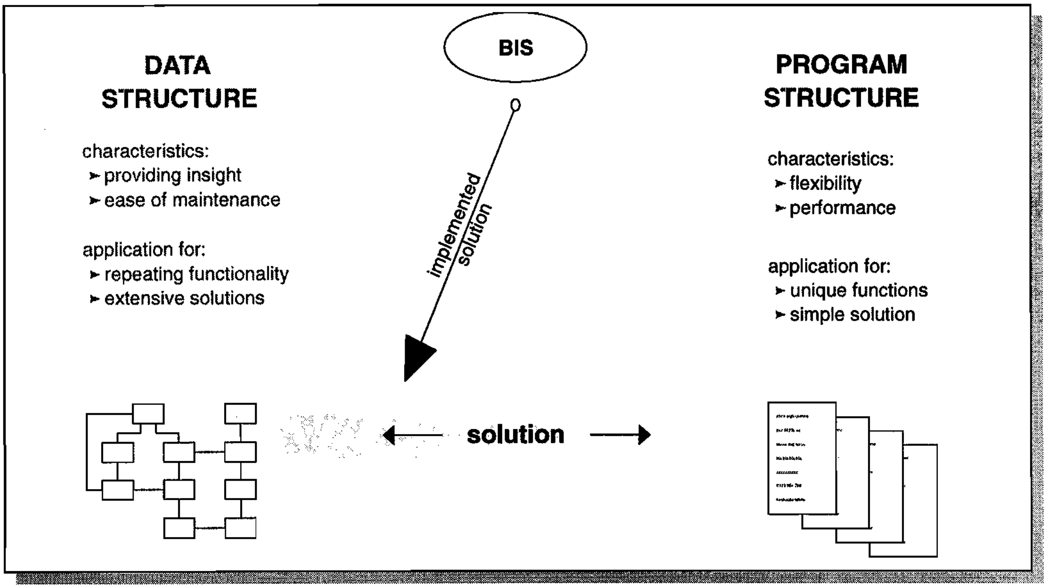


Figure 3.5 - Alternative solutions in the data structure versus the program structure

3316 Summarising, the expected insight into the functionality offered, can be provided by data structures. Nevertheless, to draw the right conclusions the factors mentioned earlier should be taken into account, especially the choices made during the implementation.

Research objective and approach

3400 The difficulties and risks of the common 'black box' approach and the opportunities suggested by assessing the data structure have resulted in the following research objective:-

Assess the value of studying data structures in the process of assessing business information systems compared to the 'black box' approach that is represented by traditional questionnaires.

This objective can be divided into three subgoals:-

(a) what are the limits of the data-oriented approach: which functional requirements can and cannot be described and assessed?

(b) where and how can this approach be applied?

(c) what is the added value in relation to current methods?

3401 Essential in the argumentation are the objectivity and reproducibility of the results originating from the two methods of software assessment. Both the expert in software selection and someone who has extensive experience with a specific software package will typically be prejudiced when assessing the functionality of certain systems. Conclusions will be based more on experience than on direct answers to questionnaires or an interpretation of data structures.

3402 To safeguard these essential requirements, the following approach was used, based on some widely available standard business information systems:-

(a) definition of the functional scope and examination of the software packages in the assessment. The functional areas chosen are 'Purchasing' and 'State-independent Manufacturing Data';
(b) analysis of the data structures in the defined areas. For this purpose a method was designed, based on a referential strategy. This strategy is described in detail in chapter 4. The resulting reference data models (RDM) have been used in the data structure assessment of different software packages. To illustrate this, projections of real software packages on the RDM are shown;
(c) determination of a typical questionnaire that covers the scope of both functional areas, to be answered by the suppliers of the software package, just as in a normal software selection process. This typical questionnaire is derived in paragraph 7200 and onwards;
(d) comparison of the conclusions from both approaches and feedback to the suppliers. This comparison is described in paragraph 7300 onwards, and includes the reasons why this comparison is particularly effective in leading to an objective result;
(e) evaluation of the results in accordance with the research objectives as formulated above. The outcome of this evaluation is described in chapter 8.

3403 The research approach and the argumentation of the detailed choices made are specified in more detail in paragraph 7100 onwards.

Part 2

Design Method and the Reference Data Model

Method for using data structures in software assessment

4001 This research should also provide a method for the assessment of data structures. The statements made about data structures should be related to some assessment approach. The direct assessment of individual data structures would seem to be an obvious approach.

4002 From paragraph 4100 onwards, assuming disadvantages in the direct assessment of data structures, in which each data structure is assessed individually, a reference strategy is defined. A suitable Reference Data Model (RDM) is central to this. The assessment method that is deduced from this will be called the RDM-method.

4003 From paragraph 4200 onwards, the objective of such a RDM is firmed. Here, an approach for the development of this RDM is elaborated with design criteria and construction operations. On the basis of these operations the distinction between the RDM and a universal model or system is clarified.

4004 From paragraph 4300 onwards, the most important characteristics of the RDM constructed in this way are summarised.

4005 Finally, paragraph 4400 and following deal with the application of the RDM as a part of the reference method. Here, some guidelines are given as to how the RDM and the projections on some BIS, in the following chapters, should be interpreted. More particularly, attention is paid to the risks of surplus functionality and complexity that immediately follow from the comparison of the data structures of BIS and the RDM.

Assessment strategy for data structures

4100 The assessment of BIS data structures will start with a request to suppliers for the data structure documentation of their BIS. Next, these data structures should be mutually compared and conclusions drawn about the differences and similarities between these BIS. Further, possible relevant functional requirements should be sought within these data structures.

4101 The success of these exercises tells us something about the possibilities for assessing BIS on the basis of data structures. Here, each of these steps is worked out and on the basis of the disadvantages observed, an alternative assessment strategy and method is proposed.

Availability of the right and suitable data structures

4102 First of all the data structures of the BIS to be assessed have to be made available in some shape or form. One important aspect is the method used for and the quality of the documentation of these models.

4103 Until a few years ago, no data structures were available in a documented form for most BIS. The most important reasons for this lay in the development tools used and the years of evolution involved for the software in question. In the original design of these systems, thought will certainly have been given to the data structures as an essential step in the design process of such extensive systems. However, since their design and initial development, these systems have grown and many functions and modules have been added year after year. This process of evolution and especially the third generation development tools used certainly did make the maintenance of documentation on the data structures mandatory.

4104 In the mean time many BIS have been redeveloped, using modern development tools like case tools, object-orientation, and relational database management systems. This has led to the fact that data structures, not only as a design document but also as the documentation of the BIS, have become an inseparable part of modern BIS.

4105 For most modern BIS, some form a documented data structure is present. This does not mean, however, that they are simply available for any interested party. The suppliers consider their data structure to be the core of the BIS and it is therefore a sensitive point concerning competition. This information is only forwarded under strict conditions (in the framework of this research there were no problems in meeting these conditions).

4106 Next to this, the quality of the documented data structure is important. Any BIS is composed of many thousands of entity types, relationship types, and a multitude of attribute types. In the available data structure there seem to be large differences in the readability and ease of comprehension. Allowing for greater understanding of this documentation is usually the first step. Guidelines for the description of complex data structures, whose objective was to create better understanding, proved necessary (see also paragraph 4200 and following concerning guidelines for the design of the RDM).

Direct assessment of data structures

4107 The obvious approach mentioned at the beginning of this chapter is based on the direct assessment of the data structures. Two sides can be distinguished:-

(a) placing the data structures from different BIS side by side and interpreting the differences;
(b) searching for the functional characteristics or the requirements in these data structures in order to be able to draw conclusions about the degree of support present.

4108 However, after making some attempts, there appear to be a number of disadvantages in such a direct method of comparing all or some of the data structures, i.e.:-

(a) the mutual differences in the terminology used by the various BIS. Some examples of related terms used in various BIS are:-
    (i) 'hourly labour' versus 'subcontracting';
    (ii) 'items', 'materials', and 'articles';
    (iii) 'forward orders', 'blanket orders', 'call-off orders', and 'standing orders';

    differences are also caused by subtle variations in the languages in which the BIS was originally developed;
(b) the different philosophies of the various BIS. A number of BIS is characterised by the philosophy of the 'production model' for the semi-process industry, versus the 'bill of material and routing structures' in the typical MRP philosophy. Though different, both cover the same ground when it comes to the definition of material requirements and capacity requirements. During assessment it is worthwhile to make statements about the degree of overlap;
(c) the multitude of data (the entity types, relationship types, and attribute types) of each BIS. As a result, it is extremely difficult to filter out the most important elements while making the comparison.

4109 Some of these problems are caused or at least reinforced by the comparison of data structures of each BIS with all other BIS, see figure 4.1. This means that for each new data structure that one wishes to compare, one is obliged to place it next to all other data structures with all their differences, philosophies, and their complexity.

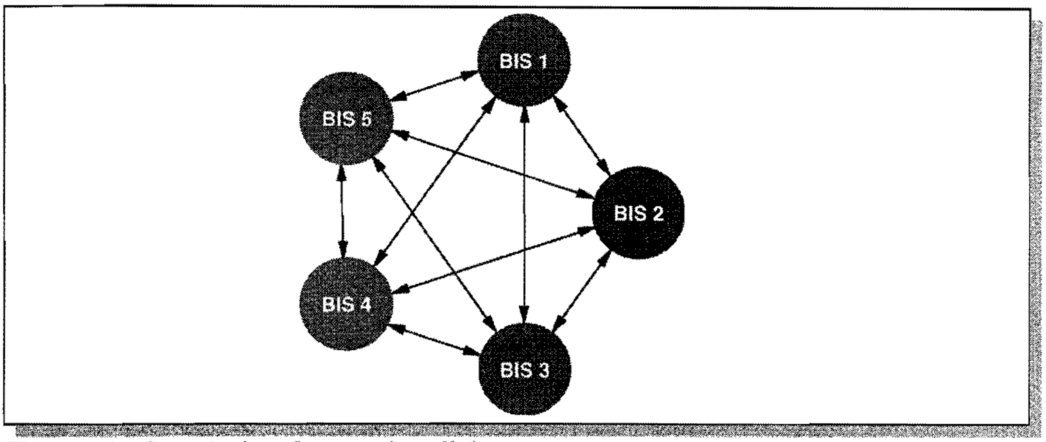


Figure 4.1 - Complexity of comparing all data structures

4110 The most important bottle-neck is the lack of a proper frame of reference to support of the process of comparison.

4111 Finding functional characteristics in the data structure of each separate BIS is problematic too. Here the most important bottle-necks appear to be:-

(a) *the functional requirements are often formulated from an entirely different perspective,* not only in terms of the company organisation but also in terms of desirable behaviour „is it possible to ....?" or „is .... supported?". In data structures this means that a search has to be made for (unknown) alternative structures that support this desirable behaviour;

(b) *a data structure of the particular company situation is not normally available.* The development of such a data structure is a considerable and complex activity that is necessary as part of the development process of a BIS, but the effort required is not proportional to the selection and use of a standard BIS. Besides, a company specific data structure need not mirror the BIS data structure, however the modelled characteristics have to be present in the BIS in some shape or form;

(c) *the data structure of the company situation is present but is either too general or too specific,* making the translation of the terminology and the structures extremely difficult. In systems that are meant to be standard, the idea is to conform to generally accepted terminology (such as APICS [API92]). In a specific company model this is not necessary and an attempt is made to conform to all characteristic features and terms of the organisation;

(d) *the functional requirements have to be translated by an expert, into possible alternative data structures and possible characteristics of these data structures* (what to look for in the data structures of the BIS). The risk exists that alternatives are skipped, and especially the alternative that is implemented in the BIS data structure.

4112 Summarising, it can be seen that several aspects are missing: a frame of reference providing insight into the terminology to be used, but also the level of detail necessary for the formulation of the required characteristics. One should avoid making choices that influence the implementation, and restrict the degrees of freedom unnecessarily while drawing up the requirements. Consider, for example, the formulation of requirements for a semi-process industry in the MRP conceptual framework, which makes the translation and checking of the original 'production model' more difficult.

Referential strategy for assessing data structures

4113 From the previous discussion it can be deduced that some kind of reference data structure can substantially improve the assessment of BIS data structures. In this research the term 'Reference Data Model' (RDM) is used.

4114 The RDM has to be a data structure that on the one hand defines a conceptual framework and on the other in which existing data structures of BIS or of business situations can be expressed. Figure 4.2 shows the effect of using such a RDM on the number of comparisons to be made with respect to figure 4.1.

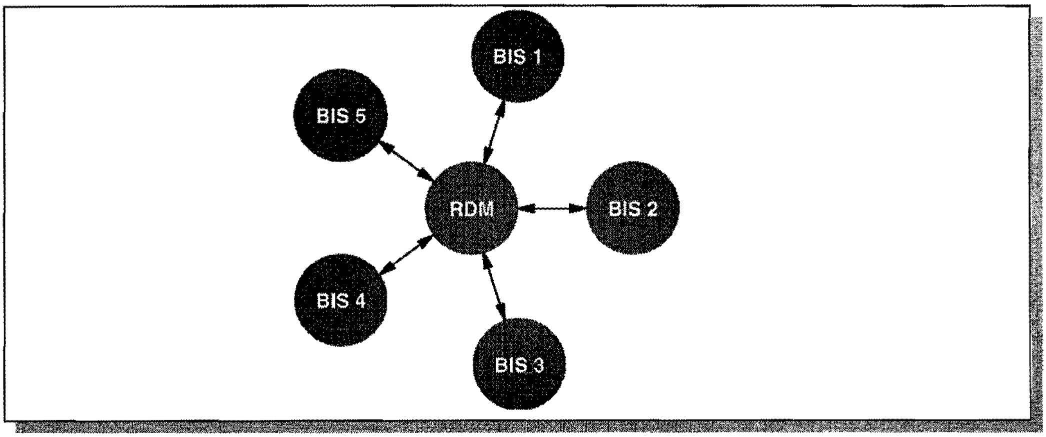


Figure 4.2 - Minimal number of comparisons using a central RDM

4115 One advantage of this approach is that a comparison of a BIS with the RDM only needs to be done once. New BIS are only compared to the RDM and the BIS previously assessed need not undergo further comparison.

4116 As terminology is defined in the RDM, choices have been made regarding the level of detail, it is relatively easy to depict a business situation or a collection of required (functional) characteristics. From this reference strategy a method can be deduced which is schematically displayed in figure 4.3.

4117 Within the method sketched, the RDM is a structure on which the BIS data structures can be projected, see the left-half of figure 4.3. After all relevant data structures have been described and structured in such a uniform way, it is possible to compare them.

4118 The right-half of this figure shows how a company specific model, which in principle has not been made in advance, is directly modelled in the terms and (alternative) structures of the RDM. The assessment then 'is nothing more' than the comparison of projections.

4119 An interesting advantage is that superfluous structures becomes visible in the comparison of these projections, since the BIS are completely projected on the RDM, including the parts that are not relevant for the specific company case. By comparing the coverage, the overlaps and the gaps are given equivalent representation. This reference method is the only approach known in which this surplus is made visible in an equivalent way to the desired functionality.

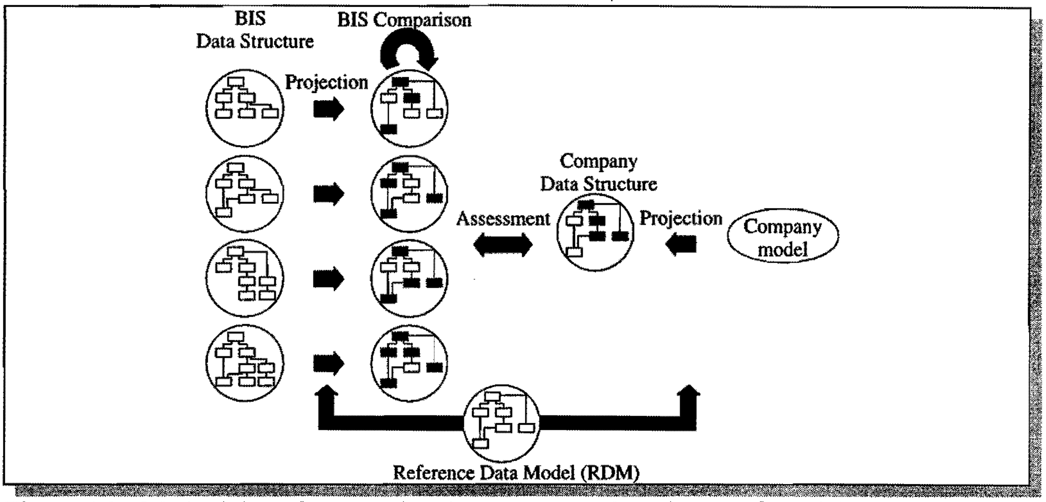


Figure 4.3 - Methodology for assessing data structures, based on a referential strategy.

4120 In the direct assessment of data structures the problem is, that in searching for the desired functional features (the user requirements), these features have to be translated by an expert into possible solutions in terms of the data structures and their characteristics. In each assessment the quality depends on the expert. Conceiving possible alternatives occurs only once in the reference strategy, i.e. in the development of the RDM. The RDM should already describe possible alternatives, thus the risk of not recognising alternatives is limited.

4121 The complexity of this RDM-method mainly lies in the development of a proper RDM and the projections of the BIS to be assessed. The comparisons of standardised data structures is relatively simple (working from the assumption that all elements to be compared are part of the RDM). The projection of the desired functional characteristics is relatively simple too. These have already been dealt with in the description of the RDM, in as far as they can be expressed in data structures.

4122 Within the framework of this research, the last step, being the projection of real company situations and their functional characteristics, is not explicitly dealt with. The research is limited to the projections of BIS on the RDM and the recovery of functional characteristics as they are incorporated in existing checklists and questionnaires. With this the research is specifically aimed at the usefulness of data structures. If the usefulness is sufficiently demonstrated, company situations can be projected on the RDM in the context of real software selections and can be compared to projections of BIS in a mechanical way as is sketched in figure 4.3. Given the proven usefulness and effectiveness of data structures, then the efficiency within data structures will receive more attention.

Development of a Reference Data Model

4200 For the development of a good Reference Data Model (RDM) the objective of this reference model will be defined bellow. This objective is the basis of the chosen technique used for modelling and describing the RDM. Next, the design and development of the RDM is elaborated on and a number of construction operations are described for combining data structures from different sources. On the basis of these construction operations, the difference between the RDM and a universal data structure is made clear. Finally, a number of pragmatic rules are given to allow this seemingly mechanical development process run efficiently and effectively, certainly considering the size and complexity of the BIS data structures.

Objectives of the RDM

4201 Models are only an abstract and simplified representation of the reality. Such a simplification is focused on the emphasis of important characteristics and the suppression of unimportant details. The recognition of this distinction between important and unimportant details is essential in drawing up a good model.

4202 The term 'reference' implies independence from a specific situation. In principle any model can be raised to the level of a 'reference model', but this does not mean that each model is equally suitable for a specific objective. Reference models have been developed from many perspectives with totally different objectives. In [GRE88] a number of different backgrounds are given from which reference models can be developed:-

(a) *for the support of a competitive strategy of enterprises.* Reference models can play a role in the competitive strategy as a means of communication with customers and suppliers, on the one hand to influence the power of negotiation and on the other to hold back the threat of competition;

(b) *to achieve greater uniformity in a given industry, branch, or complex organisation.* Here, the focus is on simplification of the communication, the strive for large-scale advantages in systems development and the prevention of confusion over terminology. Typical examples are the common systems approach[9] in complex organisations and the various projects sponsored by government for the development of standards;

(c) *bringing about an improvement in providing services or building an enhanced profile for service organisations.* Here, one should consider the extension of knowledge (for example of consultants), the development of new products (such as standard software), and achieving efficiency advantages in repeating projects.

4203 The application of reference models is extremely diverse. The concrete advantages that can be aimed for with such models (see also [HAR94]) are:-

(a) *cost reduction.* By making use of existing material one does not have to re-invent the wheel. To a lesser extent this also reduces the need for costly training and building up experience;

(b) *time saving.* Besides the development costs, development time can be considerably reduced due to the flying start one has by using existing models;

(c) *quality improvement.* By using knowledge already present and combined in reference models, use can be made of a multitude of high quality expertise. The quality is further improved through the prevention of a blinkerd view and avoidance of risks.

4204 On the basis of the above list, the following objectives for the Reference Data Model (RDM) are formulated: *The development of a data structure including an explanatory description, in which the following features must be maximised:-*

(a) *the power of definition.* The RDM should provide an unambiguous conceptual framework of terminology;

(b) *the power of projection.* The BIS to be assessed should be projectable on the RDM and expressed in terms of the RDM;

(c) *the power of discrimination.* In comparing projections of the data structures of several BIS on the RDM, relevant differences should become visible;

(d) *the power of providing insight.* Both the RDM itself and its projections not only need to be readable, they should also provide insight into the underlying philosophy, the structure and its characteristics.

---

9 In a 'common systems' approach in a complex organisation a number of common guidelines for software and hardware, customisation, and usage are drawn up and finalised with the objective to duplicate these to all different decentralised organisational units.

4205 In principle the scope of the RDM is implicitly determined by all the BIS to be assessed. In the context of this investigation, however the scope is restricted to 'Purchasing' and 'State-independent Manufacturing Data' (see chapter 7).

4206 At first it seems an obvious option to develop the RDM for a specific branch or other specific situation. However, the BIS one wishes to compare, are not restricted to one branch or industry, controlling structure or type of manufacturing. The trend is that the major BIS will support as many different business situations as possible. This means that the RDM will inevitably have to cover discrete manufacturing as well as semi-process characteristics, and engineer-to-order, make-to-order, and make-to-stock characteristics. The fact that such BIS operate successfully in such different environments also confirms the feasibility of the development of such a RDM.

> *The scope of the Reference Data Model can not be determined by a single branch or industry, controlling structure, or type of manufacturing process, as long as the business information systems are not restricted to this.*

Thesis 4.1 - The scope of the RDM

4207 The extent to which the elements of this objective can be achieved, will appear from the evaluation and conclusions in chapter 8.

4208 The objective states that the RDM has at least to cover the BIS to be projected. From this one can deduce that the RDM need not describe more than this. However, in the context of this research the opportunity is taken to go a little further into the subareas by expanding the RDM more than for the existing systems, and thus to give direction to possible future developments.

Modelling technique

4209 For describing of data structures various techniques are available. In this context a technique has to be chosen that fits the objectives of the RDM as closely as possible. Generally, one can distinguish the following descriptive techniques:-

(a) *textual description*, in which a free-format description is given in a natural language without specific rules with respect to the determination of data structures. This is the most frequently used technique both in the formulation of requirements in selection processes and in the response of the suppliers;

(b) *formal (algebraic) techniques*, like the relational model, set theory and predicate logic;

(c) *graphical (schematic) techniques*, such as Bachmann diagrams [BAC65], the Entity relationship Model [CHE76], and Nijssen Information Analysis [NIJ89].

4210 In table 4.1 these techniques have been balanced against the objectives from paragraph 4204 and by means of pluses and minuses the strength and weakness of the techniques are indicated.

4211 For the *definition of the terms* used in the RDM (these are mainly related to the existing logistics terminology, such as APICS), the obvious choice is to do this using normal text. Besides these definitions, informative descriptions and peculiarities can be given as a supplement.

	textual	formal / algebraic	graphical
Power of definition	+	-	-
Power of projection	-	+	+
Power of discrimination	-	+	+
Power of providing insight	-	-	+

Table 4.1 - Evaluation of different modelling techniques

4212 For the *projection of existing data structures* it is useful to conform to the usual BIS description methods. This means using the relational model or graphic representation. The discriminating power in graphic schematic techniques is mostly determined by the level of detail, and to a lesser extent this also applies to other formal techniques.

4213 The most important point, however, is to gain insight and understanding. Here, a graphical technique is preferential. It can be concluded that describing the RDM by using a simple graphical technique fits the objectives the best, especially when the diagrams are supplemented with textual information, definitions, and specific characteristics.

4214 Within the framework of this research it is not considered worthwhile to use a very detailed graphical technique, because it does not contribute to the RDM objective. For this reason, the customary techniques can be greatly simplified to the following symbols and rules (a detailed description of the graphical used is included in appendix B):-

(a) the modelling of the *relevant entity types* as a symbol (rectangle);
(b) the modelling of *specialisations (and generalisations) of entity types* by:-
  (i) means of a hierarchical embedding;
  (ii) connecting via relationship types;
  the choice between these equivalent types of modelling is only determined by the clarity in the specific diagrams;
(c) distinguishing *three (basic) relationship types*:-
  (i) the 1:n relationship type;
  (ii) the n:m relationship type, which should not to be normalised in the RDM as it is concise and improves insight;
  (iii) the 1:1 relationship type, usually for the description of the specialisations of entity types mentioned earlier in (b)(ii);
(d) indicating the *optional nature of relationship types* (the distinction between 1:n and 0:n) where it contributes to clarity and understanding. Further statements about the cardinality (see appendix B) of relationship types are not made. These details have no added value in the RDM;
(e) the *naming of relationships*, only in cases where this contributes to the clarity. Standard names such as 'belongs to' or 'contains' are therefore left out;
(f) indicating *exclusive relationship types*, like relationship type A **or** relationship type B;
(g) the *projecting entity types and relationship types* is shown in **bold** (where a projection can have a lower cardinality than the basic relationship type in the RDM).

4215 It is essential that the graphical technique remains subordinate to the objective of the RDM and can be applied where it is useful. The RDM is not a model that should (ever) be implemented as part of an information system.

Design of the RDM

4216 In the design of the RDM the projectability of different BIS is an essential condition. For this reason it is not advisable to develop the RDM from scratch. The projectability can be realised more easily by basing the RDM on existing BIS data structures.

4217 Another reason to use the data structures of existing BIS when developing the RDM lies in the fact that much expertise is present in these systems. Each BIS has been developed by teams consisting of tens to thousands of developers encompassing a multitude of man-years development time. Besides this enormous concentration of manpower, the existing systems are the result of a long evolution in which requirements, ideas, and experiences from many user organisations have been added. Most BIS have a long history that spans ten years or more.

4218 Though sizeable, each BIS supplier only has access to a limited amount of development capacity and a specific user group. By combining this knowledge from many of the large BIS, an even broader foundation for the RDM can be established. Considering the concentration of efforts and expertise, it would be an illusion to think one could construct this independently, resulting in the RDM. Figure 4.4 shows this 'supply-driven design strategy'.

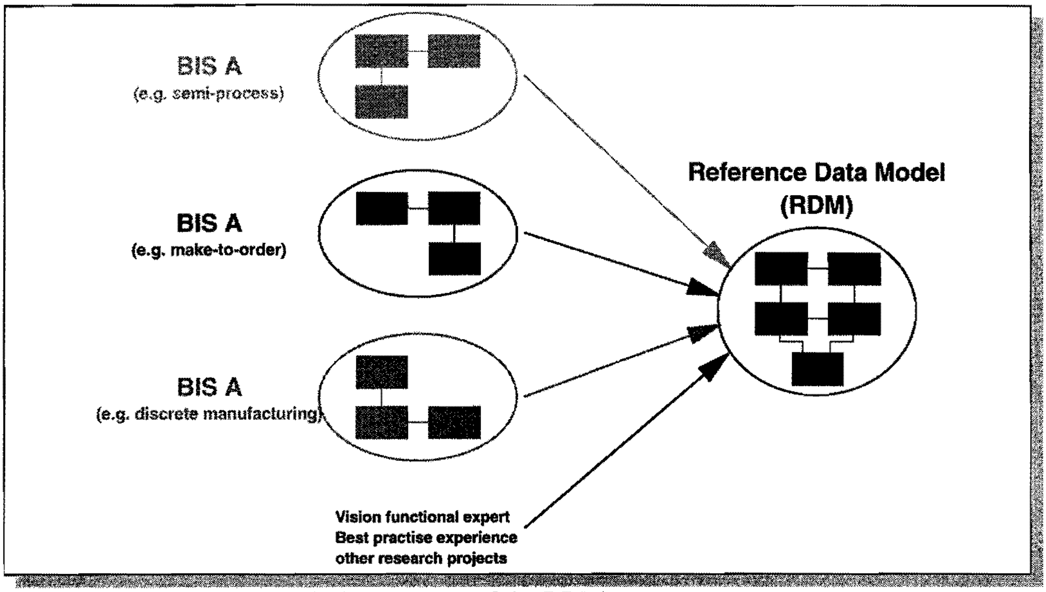


Figure 4.4 - Supply-driven design strategy of the RDM

4219 Besides this impressive combination of know-how, use was also made of advanced functional requirements, 'best practice' experiences, and recent developments in the areas concerned as supplied by various independent business experts. The development of BIS is always a reaction: it follows the requirements of the market. From the perspective of business consultancy these developments are pushed forwards. The development of new logistics controlling philosophies, for instance, and the execution of revolutionary Business Process Re-engineering also results in entirely new functional requirements. It is not the RDM's objective nor does it pretend to cover or guide all these new developments, but the addition of these elements does lend extra added value to the RDM.

4220 The development of the RDM makes use of a number of renowned BIS as its starting point, among others SAP-R/3 by SAP, Triton by Baan, Prism by Marcam, Ratio by JBA/Rationplan, TXbase by TXbase systems, and Prodin-P2 by Inter Access Infostore.

4221 During the elapsed time of this research, the RDM has been continually extended, particularly through the addition of new BIS. The RDM is clearly viewed as a growth model. While projecting data structures from new BIS additions to the RDM will be made, as has actually occurred in recent years.

4222 Therefore the result of the development process is a RDM that confirms the present state of the art regarding the knowledge and experience in the functional areas concerned.

Construction operations: the RDM compared with the Universal Data Model

4223 Basically, the RDM is constructed by combining or joining existing BIS data structures. This may seem to result in a kind of universal data model (UDM): the data model of a new BIS that works for every situation. However, with respect to the objectives of the RDM mentioned earlier, the objective of such a UDM clearly differs. *The UDM can be defined as „A data model as part of an information system that generally applies to all business situations within a given scope*". This scope can be defined as: a branch (for example machine building industry), a type of controlling structure (for example engineer-to-order), or more generally 'all industrial manufacturers'. The result of the latter scope provides a data model that would be suitable for any type of manufacturing organisation, something that a number of BIS suppliers indeed are striving for and working on!

4224 *The RDM on the other hand is not intended to be a universal model but a comparative model.* There are a number of essential differences between the UDM and RDM. These differences become clearer in the description of a number of (principle) construction operations for both models. Below the most important construction operations are described:

Extension operation

4225 The extension operation is a basic operation for the extension or addition of data structures and is used by both the UDM and RDM.

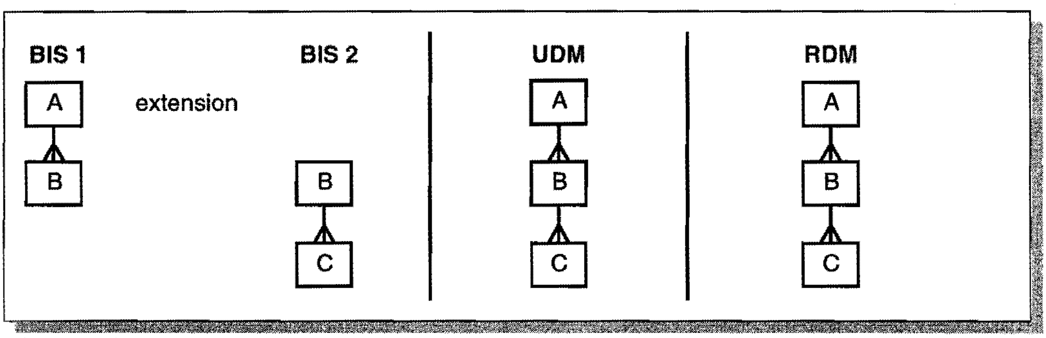


Figure 4.5 - Extension operation

Combination operation

4226 The combination operation is a comparably basic operation that links data structures. This also is used in the construction of both RDM and UDM.

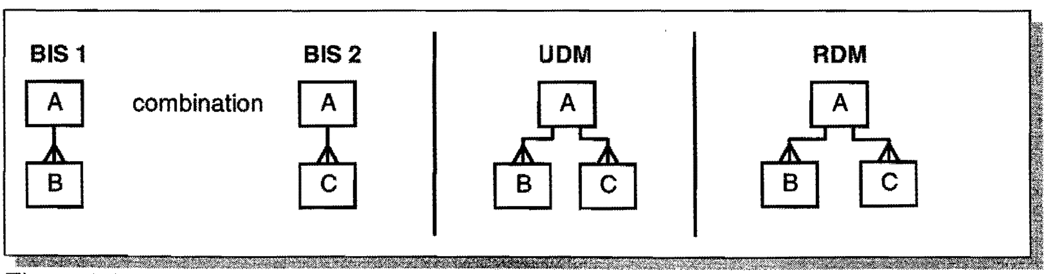


Figure 4.6 - Combination operation

Selection operation

4227 The selection operation leads to basically different results in the UDM and RDM. If the data structures of two BIS are combined and both contain alternative realisations in the data structure for comparable functionality, the UDM has to make a choice (choose the best possible). However, in the RDM the fundamental objective is to make alternative solutions visible. As a consequence, the RDM must therefore contain all possible (alternative) solutions (only as far as they can occur in reality).

4228 One example is the 'production model' versus the 'bill of material and routings'. In the RDM these different philosophies have to be positioned with respect to each other, only then is the projection of different BIS possible. In contrast, in the UDM an explicit choice has to be made for one of these philosophies. If no choice is made this would lead to an enormous number of superfluous entity types, and particular relationship types, mostly in the complex form n:m. The disadvantages of this are further explained in section 4418.

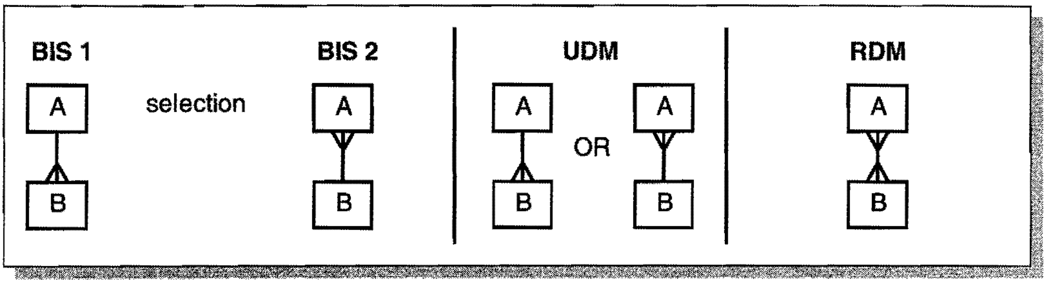


Figure 4.7 - Selection operation

4229 In figure 4.7 this is expressed by the alternative relationship types between A and B. Instead of accidentally introducing unnecessary complexity, like the complex n:m relationship type or even contradictions, principle choices have to be made in the UDM. That does not mean to say that no n:m relationship types are possible at a logical level in the UDM, but they have to be logically consistent. Fundamentally this is not the case in the RDM, and simply combining the two relationship types is allowed without further bothering about consistency or contradictions. The only restriction is that relationship types with different meanings should be shown separately and not 'added up'.

Specialisation operation

4230 Specialisation is a type of structure that occurs in nearly every data structure. Nevertheless almost every designer or developer makes different choices in the level and aspect of specialisation. Thus, one model may distinguish an entity type called 'Resource', which is specialised in consumable and durable resources, while another model may have separate entity types for materials, machines, personnel and tools. A number of choices lie at the foundation of such specialisations, which are related to the area of application (functional specialisation) and to implementation considerations (technical specialisation).

4231 In the UDM an attempt must be made to achieve as universal a data structure as possible, which usually means that generalisations are preferred to functional specialisations, as they are more suitable for a multitude of application areas. For instance, in the sales process there are many types of orders (forward orders, project orders, commission orders, etc.) that are defined as generalised entity types in the UDM, based on large numbers of parameters for different treatment and control. In figure 4.8, B1 and B2 are functional specialisations of B, which are generalised in the UDM.

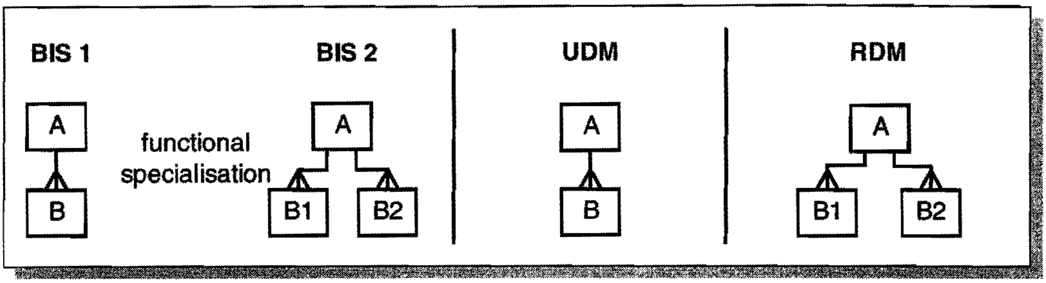


Figure 4.8 - Functional specialisation operation

4232 It is clear that for *technical reasons* specialisations are introduced to facilitate implementation, for example a separate view of an item master file of different departments (sales, purchasing, warehousing). Figure 4.9 shows the result of this technical specialisation in the UDM.

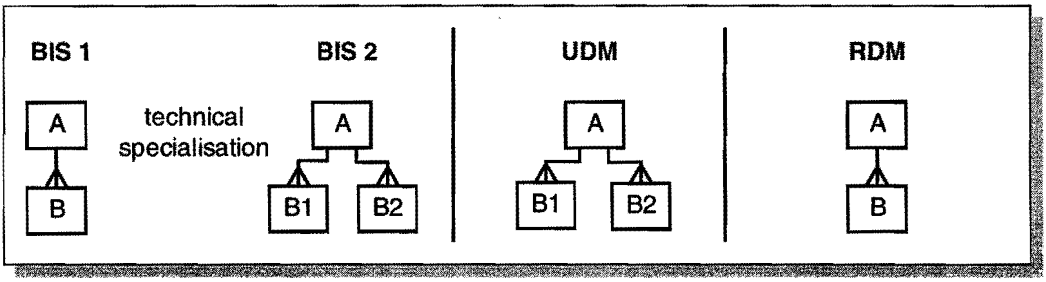


Figure 4.9 - Technical specialisation operation

4233 However, in the RDM the functional distinction between the various specialisations is relevant and these are normalised wherever possible, see figure 4.8. The consequences for implementation are totally irrelevant, which implies that nothing is demonstrated by technical specialisations, see figure 4.9.

Redundancy operation

4234 Redundancy quickly arises when data structures are combined. In the UDM this redundancy is usually undesirable. Due to the need for data consistency, unambiguous choices have to be made in the UDM. Besides, redundancy often means additional data entry, extra handling and superfluous transactions for users.

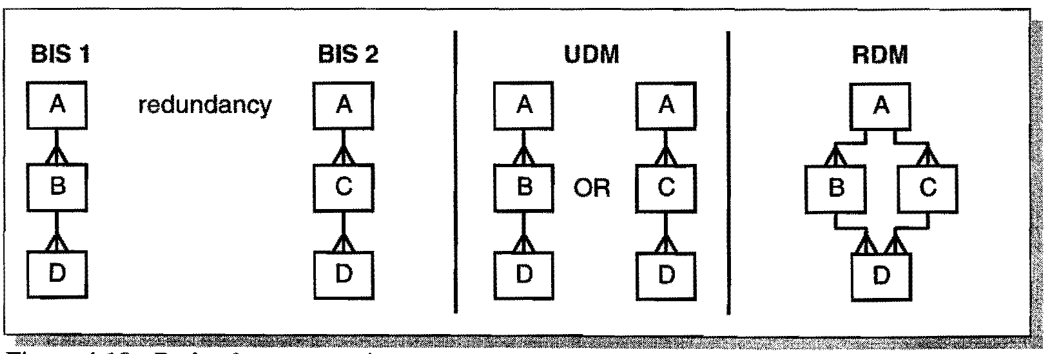


Figure 4.10 - Redundancy operation

4235 In the RDM this redundancy is essential to be able to position the various alternatives with regard to each other and to compare them.

Generic operation

4236 In the analysis of data structures, and certainly in the combination of data structures different sources, certain constructions appear to occur in several places. One example concerns alternatives for materials, bills of material, routings, and operations. Each time we see a repetitive structure, which can be found in numerous places. In some places such a data structure can be meaningful, while in other places it is more theoretical.

4237 For example, alternative operations should be modelled in the data structure, while alternative machines or labour might not be modelled in the data structure. Instead these could be solved by means of alternative operations or machine groups and labour groups.

4238 In the development of the UDM existing data structures are combined. The coincidental occurrence of alternatives in the data structures to be combined are included in the UDM (for example alternative materials and alternative operations).

4239 In the construction of the RDM such repetitive data structures are recognised. Besides combining the existing data structures, this recognised structure is added in other places where this had not been the case in the BIS under consideration. Thus, the alternative machines mentioned earlier will be included, despite the fact that they were not found in the BIS.

4240 Due to this generic extension, the RDM already takes advantage of data structures that otherwise would be added at a much later stage. More importantly, this generic extension provides additional insight into functionality that is not covered or supported by the projected BIS.

Detailing operation

4241 Certain attribute types or relationship types can occur at several hierarchical levels in a data structure. Thus, a 'discount percentage' can be determined per order line, per order, but also per supplier. The same applies, for instance, to the relationship type of each of these entity types to 'Ship From Address'.

4242 In the UDM all occurring attribute types and relationship types have to be modelled. In the RDM this would lead to a cluttered structure that adds little information. In view of the objective of the RDM, it is sufficient to determine this data at the lowest level and to display the basic information.

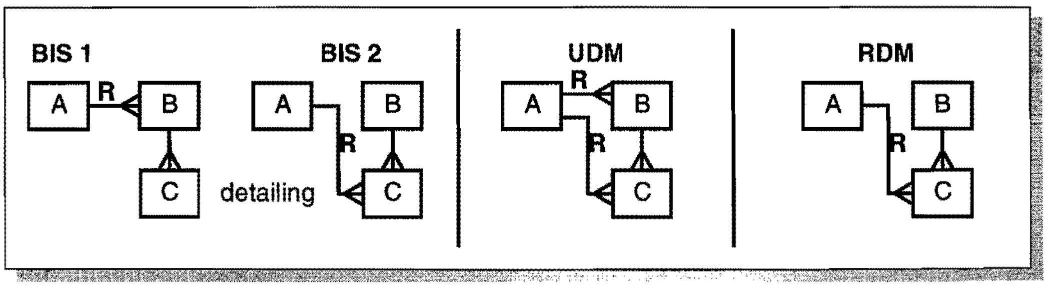


Figure 4.11 - Detailing operation

4243 Naturally, the levels supported do make a difference to the functionality of the BIS. This means that the relationship types will have to be displayed at the right level of detail in the data structure projection.

4244 To recapitulate, the following thesis can be formulated for these construction operations for the RDM in contrast to the UDM:

> *The Reference Data Model is not a model of a universal Business Information System (BIS) but can be used for the generation of specific information systems.*

Thesis 4.2 - The difference between the RDM and a universal BIS

4245 This is caused by the divergent design objectives, as well as the intentional introduction of redundancy and inconsistency in the RDM. However, the expertise stored in the RDM can also be used in the development of a BIS.

4246 The construction operations described above are very formal. During the development of the RDM, the objective should be kept in mind. Otherwise, this could lead to (documented) adaptations and to extra relationship types or generalisations/specialisations being added.

Pragmatic rules for constructing the RDM

4247 Besides the formal construction operations described earlier, a number of practical rules can be given for developing the RDM. These rules cannot be entirely unambiguously defined and some freedom with regard to the construction process in favour of the objective is desirable. So, too, this process should not be pinned down by the constraints of the schematic techniques, development tools and drawing tools used (for example case tools place demands on the consistency, which in this case is undesirable). Most rules are therefore based on experience gained from this research and the development of the RDM.

4248 Some simple rules:-

(a) in drawing data structures the readability is greatly enhanced if a hierarchical structure can be shown from top to bottom. For example: the hierarchy of supplier, purchase order, purchase order line, and receipt. Though this may seem obvious, suppliers often appear not to have documented their data structures in this way;

(b) the n:m relationship types are not normalised, in fact they are introduced wherever possible. Normalisation is only necessary for the implementation of data structures, which does not play a role in the RDM. It is more important to keep the RDM compact. Anything that contributes to compactness should be considered, certainly if this can be brought about without loss of information while maintaining the n:m relationship types;

(c) attribute types contain much information but are not modelled in the RDM! The most important reason is the enormous quantity of possible attribute types, being a multiple of the entity types. Readability and comprehension would be totally negated if these were to be included in some way. In the accompanying descriptive text some important examples will be mentioned. To trace the functionality in attribute types one is referred to the original data structure of the BIS concerned.

4249 We are still left with a large number of entity types and relationship types. To use them selectively a typology has been made. This is an aid in limiting the entity types that add little or no information. Figure 4.12 shows this typology as a hierarchical structure.

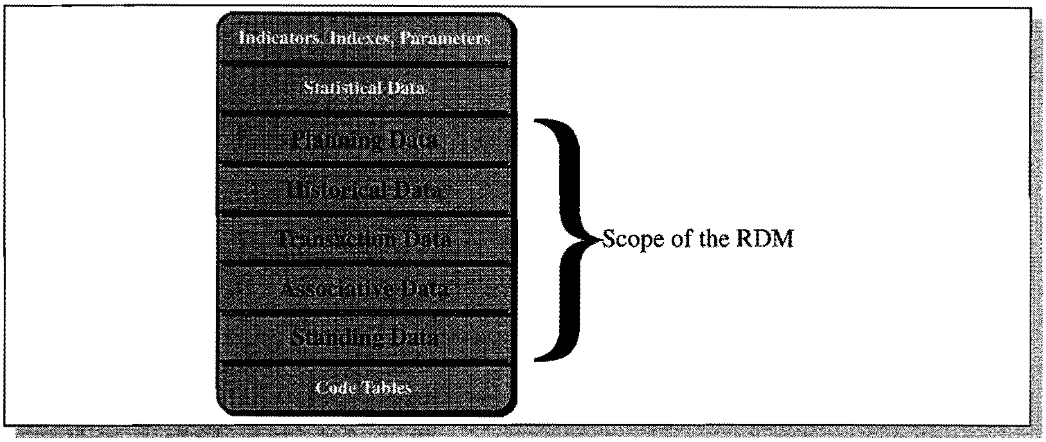


Figure 4.12 - Typology of data structures and entity types

4250 In this typology one can distinguish the following levels, bottom up:

Code tables

4251 Code tables contain all singular coded information used in the system and often consist of only an alphanumeric code and its textual description. Typical examples are country tables, foreign currency tables, item groups, and branches. These code tables are not normally relevant for the evaluation of BIS. Keeping these outside the scope of the RDM means a major simplification of the size of the data structures. In special cases fundamental functions lie behind code tables, for instance, foreign currency tables or (item) codes with classification and search structures. In these cases, special attention has to be paid to these subjects separately, in the RDM (and not to all the data structures where reference is made to these code tables).

Standing data

4252 The standing data consists only of the basic entity types that can be directly and physically found in a certain company. Typical examples are customers, machines, products, departments, and suppliers. The presence of these entity types is usually so obvious that it provides no useful discrimination between BIS. However, in their attribute types large differences can occur. These entity types are included in the RDM because the data structure of any system (as well as the RDM) is based on them.

Associative data

4253 The associative data also consists of static (*state-independent*) entity types, formed by the diversity of relationships between the various standing data and between standing data and other associative entity types. Typical examples are bills of material, routings, alternative items, and price lists. Here, large differences can certainly be expected between the various BIS and therefore they are extremely relevant for the RDM.

Transaction data

4254 The transaction data consists of *state-dependent entity types* that are basically the dynamic derivations of the associative entity types extended by the factor of time. Typical examples are quotations, orders and order lines, price agreements, and stock movements. Here comparable differences between the various BIS can be expected as with the associative entity types, as they are direct derivations. These entity types are also included in the RDM.

Historical data

4255 The historical data contains older versions of (mainly) transaction data, in so far as these are considered relevant. The entity types are used on the one hand to trace data and are used on the other in reporting. Typical examples are price history, order history, logging and audit trails. As historical data is by definition a derivation of the transaction data (and possibly from some lower level, state-independent, entity types), it is of subordinate importance in the evaluation of the BIS. In this research it is expected that these entity types will not play an important role. They will only be partially included in the RDM and are mainly treated generically.

Planning data

4256 The planning data consists of the entity types used at the tactical control level of a company. Typical examples are the master production plan, simulation plans, and budgeting. These entity types can be expected to be clearly distinguished in the different BIS, but the arithmetic rules applied in the programming structure are also important in the comparison. Data structures can only be expected to provide part of the desired insight. For the relevant functional areas the entity types are included in the RDM.

Statistical data

4257 The statistical data consists of the entity types that are only intended for the purposes of reporting, mainly where a basic structure is known in advance and the historical data becomes too bulky to provide the information. Typical examples are sales statistics, supplier performance, and monthly material usage. This data is of course important for controlling a company. However, in the modern BIS various tools are included to define such statistical files at implementation. Therefore, the selection criterion is the presence of such tools, rather than the presence of pre-defined entity types. In the RDM these entity types can usually be left outside the scope.

Indicators, indexes and standards and parameters

4258 The indicators, indexes standards and parameters are more individual attribute types that are necessary for algorithmic decision support. Where possible these attribute types have been added to existing entity types. However, in stand-alone (decision-support) systems the data groups do form complete data structures. Typical examples are performance indicators for delivery reliability and quality and control parameters for stock policies or demand forecasting. The implicit functionality is expected to lie within the software coding. No more than an indication can be expected from the presence of the data.

4259 In the specific delimitation of the scope of the areas 'Purchasing' and 'State-independent Manufacturing Data', we shall take a closer look at some specific choices within this typology.

Characteristics of a Reference Data Model

4300 In the previous part of this chapter the construction of the RDM was discussed in detail. Here, a brief characterisation of the result of this construction process will be given. As stated the RDM is certainly not a data structure for a new BIS, where other design criteria apply like effective and efficient processing of data and unambiguous and consistent structure. In contrast the RDM combines a number of contradictory concepts and these alternative structures and solutions conflict with the consistency normally required in the data structure. Furthermore, the RDM is far from complete as a data structure for a BIS, among other things due to the selective inclusion of entity types and relationship types.

4301 The RDM should be a frame of reference for a diversity of BIS that manifest themselves in a multitude of company situations. Whereas the data structures of BIS each have their own language, terminology and structure, the RDM can be considered to be a meta-language (an Esperanto) that can be used to translate individual data structures. In this sense the RDM can be viewed as a meta-data structure of the BIS data structures.

4302 Other characteristics of the RDM are:-
(a) the RDM is the smallest common multiple of BIS data structures;
(b) the RDM contains the concentration of knowledge that has been built up independently by suppliers;
(c) the RDM provides a structure in which this knowledge and these experiences can be stored.

4303 Furthermore the development of the RDM is by definition never completed because BIS will always continue developing.

4304 Finally, in figure 4.13, a number of frequently occurring characteristic structures in the RDM are shown.

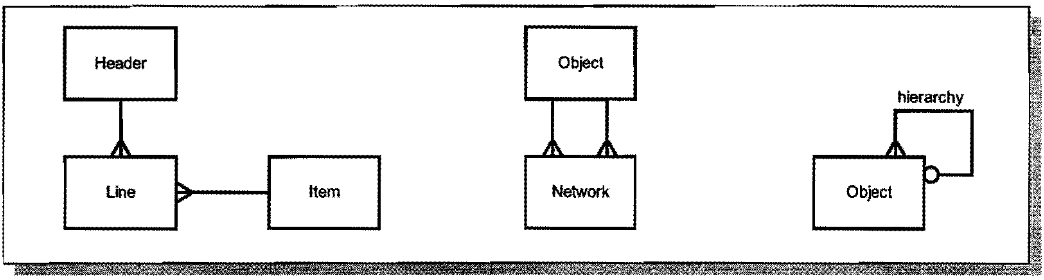


Figure 4.13 - Characteristic RDM data structures

4305 From left to right we see the following in this figure:-
(a) *the line structure* in which multiple lines are linked to one header. Examples are a (purchase) order and (purchase) order lines, a bill of material and bill of material lines, a shipment and shipment lines. Each line contains an individual item (materials, item or resources) and they are clustered at the header level;
(b) *the network structure* in which the n:m relationship of an entity type towards itself is normalised by means of a supporting entity type. Examples are:-
  (i) alternative materials - several alternatives are possible for one material, but one material can also be an alternative for several other materials;
  (ii) bill of material structures - the traditional parent-component relationship;
  (iii) sequencing of operations - the start-start relationships and end-start relationships of a network of operations;

(c) *the hierarchical structure* in which an entity type referring to itself defines a tree structure. Examples are a (supplier) organisation structure and a classification hierarchy of items. This is therefore a simplification of the network structure.

4306 These three characteristic structures can be found in many places in the RDM.

Application of a Reference Data Model

4400 In chapters 5 and 6 the RDM is described for two different functional areas. For illustration purposes a number of typical examples of real BIS data structures have been projected. The objective of the RDM is to make statements about these data structures and projections.

4401 Here, a number of useful rules are given for the projection of BIS on the RDM, which basically tell us how to use the RDM. After a projection has been performed, the next step is to interpret these projections. This will be discussed in detail in chapters 7 and 8. In this context it is useful to make some general notes before studying the RDM by means of the illustrative projections. Finally, the pros and cons of surplus functionality and complexity will be discussed.

Rules for projecting data structures on the RDM

4402 The basis for the projection is the search for corresponding entity types in the RDM and in the data structure to be projected. If an entity type proves to be present in the data structure of the BIS then it is projected (see entity type B in figure 4.14). Projection also takes place if an entity type from the RDM is found as part of a more extensive entity type. Here, the RDM is more specialised. The condition is that the BIS entity type consists of more than one logical attribute type, if actual coverage is to be claimed. If one is dealing with only one attribute type in the BIS data structure, then only the frame of the entity type concerned is indicated **bold** in the RDM (displayed as entity type C in figure 4.14). This implies that the entity type is present at a logical level, but can not contain any characteristics or attribute types of its own.

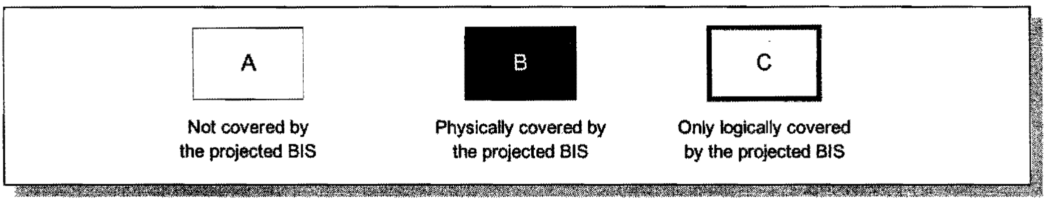


Figure 4.14 - Difference in projection of physical and logical entity types

4403 As far as the projection of relationship types is concerned, a comparable method is used to see whether they can be found in the data structure of the BIS in question. Here, we should not only look for the physically implemented relationship types but also for the hidden relationships in attribute types. Particularly in relational database management systems, it is possible to lay ad hoc relationships between entity types through a so-called 'join' operation. Such a 'join' can create a relationship between the entity types 'Purchase Order Line' and 'Sales Order Line' on the basis of the occurrence of an attribute type 'purchase order' in the 'Sales Order Line'. Though the relationship is not explicitly defined in the data structure, one can assume that it is definitely present.

4404 This means that the presence of attribute types with the same definition in two entity types indicates the existence of a logical relationship. In the projection of this particular BIS on the RDM such a relationship type will therefore be projected. If such an attribute type is a unique identification or key of the two entity types, it implies a complex relationship of the n:m type, which are not normalised in the RDM.

4405 As has been described in paragraph 4241, relationship types have only been included in the RDM for the lowest level of detail. If this relationship type is present in the data structure to be projected, but at a higher level, then it will be added to the projection.

4406 Then there is the special case of the so-called 'repeating groups' in data structures. These concern one or a group of attribute types that are repeated several times within one entity type. A typical example is 'volume discounts' that are often implemented as attribute types and repeated ten to fifteen times in the entity type 'Purchasing Condition'. Such a 'repeating group' can immediately be projected onto the normalised entity type 'Volume Discounts' in the RDM. This contrasts with hard coded attribute types that have a fixed meaning like for example 'pallet discount'.

4407 Another example of a hard coded repeating group is the additional costs of a purchase order, such as transport costs, custom costs and packaging costs. These specific attribute types do not justify the projection on the normalised entity type 'Additional Purchasing Costs'.

Interpretation notes of projections on the RDM

4408 In the interpretation of projections on the RDM, it is essential that one realises that if more is covered (projected) this does **not** mean that more is possible (or that it is a better BIS)!

4409 Firstly, alternative solutions in the RDM have been modelled that are unnecessary and undesirable as a combination in one BIS. The RDM says nothing about which alternative solution is best. In the additional information about the RDM something is said about the possibilities and limitations of these alternatives, but this could be either a disadvantage or an advantage in any specific situation.

4410 Secondly, the RDM only describes the most detailed relationship types (see detailing operation in figure 4.11 and paragraph 4241). However, this does not mean that projections on these detailed relationship types are better than at a higher level. For instance, consider delivery addresses which are determined at the lowest detail level: the purchase order line. In practice it could be obvious that such an address is often the same, certainly if one only has one warehouse, or that it only differs per purchase order. It would be far too much work to enter the delivery address for each purchase order line (besides the fact that the error rate will increase). So in a single site situation, a delivery address per purchase order is much better than per purchase order line, while in certain multi-site environments (with a centralised purchasing function) the reverse is true.

Risks of superfluous functionality and complexity

4411 A higher degree of coverage of projections on the RDM might mean a more extensive system with more complexity. Though it is theoretically possible to hide all this complexity (for example by pre-configured versions of the BIS), the present generation of BIS has not yet fully succeeded in this.

4412 Besides large and complex multi-national companies, there is large group of medium-sized companies interested in fast and cheap systems. There may be a need for broader functionality but not for all the complexity. However, which functions are or are not necessary differs from company to company.

4413 If surplus functions are not hidden then this will be expressed directly in the following ways:-

(a) the number of menu functions that contain all these functions;
(b) the number of screens, for example those necessary for the input of a sales order;
(c) the enormous quantity of data fields on the various input screens. Many hundreds of data fields may be required for the input of a single item;
(d) the number of steps or key strokes that a user has to carry out for one (simple) transaction;
(e) the complexity of the data structure;
(f) the initial costs of among other things the software and hardware.

4414 These need not be just disadvantages, because through organisational or technical measures a number of potential disadvantages can be removed. Besides that, some aspects may only appear to be disadvantageous, while more important ones are, for example, hidden in the implementation or use of the system.

4415 The potential disadvantages and possible solutions of information systems that are (too?) large and complex can be:-

(a) *the higher purchase costs of the software.* This can be compensated by a larger market share of the broader applicable software so that prices may remain low. In principle, the sales price is determined by market forces and not by the complexity of the software;
(b) *the increased hardware requirements.* The argument is often used that hardware is getting cheaper and cheaper, which is certainly true. However, the requirements made by the software also keep increasing;
(c) *the amount of time one needs to install and configure a BIS.* Experience teaches us that we are not concerned with the direct effort in setting the parameters, but mainly with the time needed to discover the consequence of each parameter setting in such a complex system. From this, we can deduce that the time for parameter settings can be controlled by people highly qualified for the job. Besides that there are a number of technical developments taking place in the area of pre-set versions (or templates), branch models, and advanced tools that help reduce this effort (and implies less dependence on a small group of experts);
(d) *the extended training period for users.* Besides paying more attention to the efficiency of the training process, more and more use is being made of multi-media tools. As a result training does not need to take place in a traditional class room, but can be followed at a time and place suiting each trainee;
(e) *the higher maintenance costs*, often expressed in a percentage of the purchase price. Here, the same considerations apply as for the purchase costs of the software;
(f) *the risk of unprofessional practices during implementation of non-essential functions.* In simple systems this risk does not exist. However, in the implementation of an enterprise-wide information system, thought should be given to new working practices that make optimal use of the BIS. Optimal also means using only the functionality that really provide some advantage. Besides 'designing' this optimal way of working, which is of obvious importance for all implementation processes, guarantees should also be made by means of project structuring and project management to ensure actual achievement;

(g) *modifications are far more complex and therefore more expensive.* On the other hand fewer modifications are necessary in such extensive systems. Modifications are undesirable for the type of companies mentioned. Moving along with the developments of the BIS supplier is normally far more important, working on the assumption that the BIS is aimed at the branch or market segment to which the company belongs;

(h) *the additional actions in operational use.* The only way one can avoid this is by means of tools that make these steps or key strokes invisible or can automate them before or during the configuration of the BIS.

4416 From this we can deduce that the assertion that '*complexity costs money*' cannot be defended immediately. A series of extra requirements and conditions do follow from this, especially concerning critical expertise. This implies that in equal suitability a simpler BIS is always to be preferred. The problem lies in the definition of equal suitability. A more extensive BIS also offers advantages, particularly in a changing company situation and to cater for unexpected future developments.

4417 For the interpretation of projections on the RDM, surplus of data structures can be recognised with different levels of disadvantage. Figure 4.15 gives four projections in which the desired situation is projected in *bold-black* with respect to the data structure of the BIS being assessed.

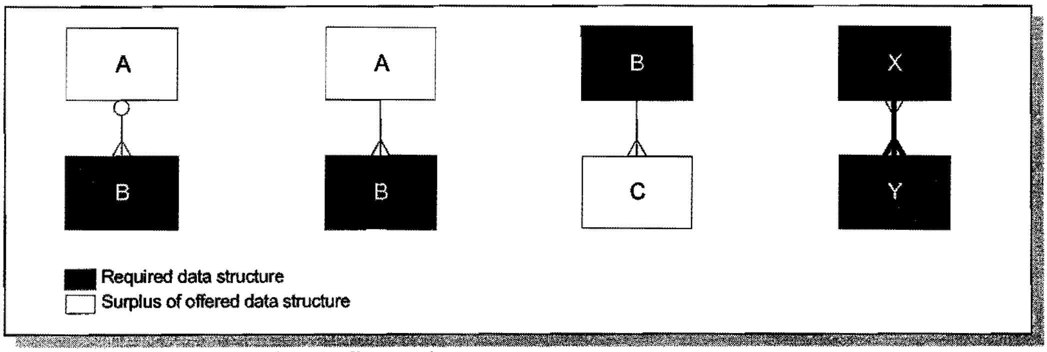


Figure 4.15 - Samples of superfluous data structures

4418 In the assessment of a surplus it is important to ask the question how much trouble the surplus will cause. With regard to the four example projections one could apply the following:-

(a) *the relationship type with 'A' is optional.* This means that the relationship can be created during the input of an entity of the type 'B'. Normally, this can also be skipped without any difficulty. The possible burden of the surplus is mostly limited to *skipping a data field in the input screen*;

(b) *the relationship type 'A' is mandatory.* In contrast with the previous structure it is mandatory to make reference to an entity type 'A'. Besides creating a relationship to a possible dummy, it must be realised that many systems may link far more functionality to this sort of mandatory relationships (otherwise is would have been optional);

(c) *the entity type 'B' has no relationship with a third entity type 'C'.* This means that there is a separate input screen that is linked to the input of 'B', which can often be ignored;

(d) *instead of the complex n:m relationship type between 'X' and 'Y' a simple 1:n relationship type is required.* A complex relationship type is normally implemented as an extra table. This means that its maintenance may entail a great many extra actions, which may be unnecessary in a simpler structure.

4419 These are only anonymous examples of the possible consequences that a data structure too complex for the requirements can have, concentrating on the data entry. However, in reporting and queries one can also be hampered by the superfluous data structure. It is important to keep this in mind when assessing the projections on the RDM that maximum coverage should not be strived for. It is important to have some idea of the consequences of having a superfluous data structure.

An effective Reference Data Model for Purchasing

5001 Before setting up a Reference Data Model (RDM) for purchasing, the scope of this functional area is defined in paragraph 5100 and following. An outline in terms of activities is deduced from this extensive functional definition of purchasing. Next, a summery is given of the different types of purchasing groups.

5002 From paragraph 5200 onwards the actual RDM for purchasing is worked out in detail. For this purpose the entire model is built up step by step in a sequence of subjects. Also, in every step the data structure of a real BIS is projected onto this RDM. This gives an impression of the degree to which projections distinguish themselves from the entire RDM, and of the functionality that is covered by a more or less arbitrary chosen BIS.

5003 Due to the diversity of purchasing groups and their relation with the environment (sales, production etc.), these flows and purchasing situations have been worked out in the RDM only generically. From paragraph 5400 on some of these frequently occurring purchasing situations are worked out in more detail from this generic model.

5004 Finally, from paragraph 5500 onwards a summary is given of the discriminating features of purchasing systems and particular those that can be deduced from the RDM.

Scope of the functional area of Purchasing

5100 Purchasing as a profession and specialisation is developing rapidly and is very much in the limelight at present. In the mean time an increasing amount of (scientific) theory and literature is becoming available. For the definition of this area particular use was made of a (Dutch) standard work in this area 'Purchasing from a strategical perspective: analysis, planning and practice' [WEE88]. Furthermore, extensive use was made of the knowledge that can be found in a large number of specialised purchasing systems and BIS, as well as the practice of consultancy in this area.

5101 The foundation can be formed by a functional definition of the term purchasing: „*The procurement from external sources of all goods and services that are necessary to run a business, for maintenance and for management under the best possible conditions for the organisation*" [WEE88].

5102 It should be noted here that there is a difference between purchasing as a function and the role played by a purchasing department. The purchasing department is not always involved in the complete purchasing process, and certain types of purchasing take place outside the purchasing department altogether. Some examples of procurement where the purchasing department is often not involved, are:-

(a) special services such as the engagement of consultants, cleaning services, an accountant or negotiating insurance;

(b) sizeable investment projects like the purchase of a new office block or the selection and implementation of standard software.

5103 For these more exceptional cases, the main role of the purchasing function lies with the (senior) management or a staff department for example.

5104 The 'external source' in the definition implies a certain degree of independence between purchasing company and the supplier. The organisation has (some) freedom about the choice of suppliers and even if a supplier has a monopoly in the market there still is room for negotiation. The suppliers in this context are considered to be (other) legal entities to which financial obligations are due after procurement. This distinguishes them from the provision of material requirements by deliveries from other sites of the same company; here there is no financial influence therefore Distribution Requirements planning (DRP) for example falls outside this definition of purchasing.

5105 A further outline of purchasing based on this definition can be defined out from three perspectives:-
(a) the specific purchasing activities;
(b) a specification of the purchasing groups under consideration;
(c) a typology of data elements and data structures in information systems.

Specific purchasing activities

5106 The analysis of the complete purchasing process provides a series of activities that are summarised in figure 5.1. Here a distinction can be made between tactical and operational purchasing

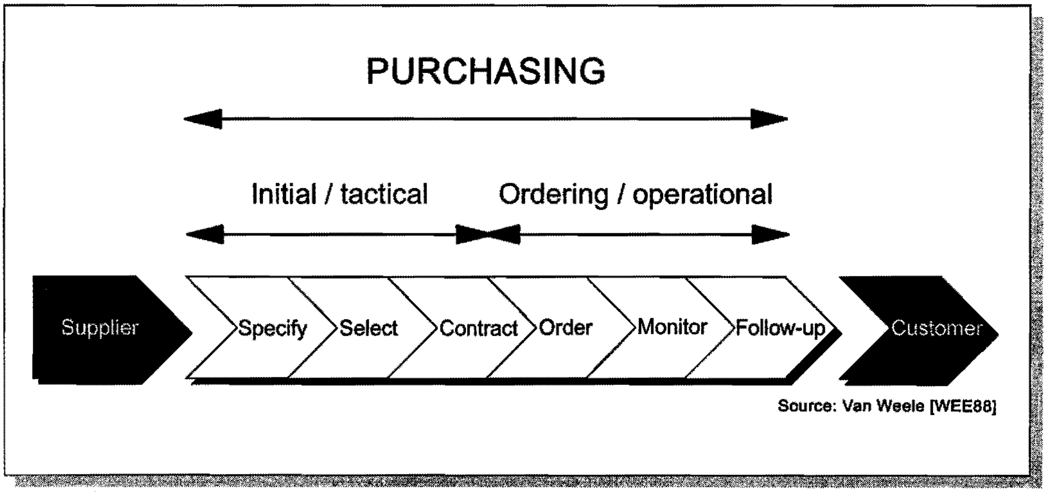


Figure 5.1 - Overview of purchasing activities

5107 At the tactical level, also known as initial purchasing, the following activities are important:-
(a) *specification*:-
  (i) identifying the problem area;
  (ii) determining the (technical) specifications (qualitative aspects);
  (iii) determining the requirements in a quantitative sense;
(b) *selection*:-
  (i) searching for and selecting potential suppliers;
  (ii) requesting quotations;
  (iii) evaluating the quotations;
  (iv) negotiating with suppliers;
(c) *contracting*:-
  (i) choosing the definite suppliers;
  (ii) agreeing contracts.

5108 At the operational level, the actual procurement and settlement, focuses on the following activities:-

(a) *ordering*:-
    (i) calling off or ordering goods or services;
(b) *monitoring and settlement*:-
    (i) monitoring the progress of orders;
    (ii) logistical settlement (such as receipt, quality check, and returns);
    (iii) financial settlement (such as registration, checking, and payment);
(c) *follow-up*:-
    (i) evaluation and feedback (vendor rating);
    (ii) finalising (such as settlement of claims and archiving).

5109 The support of both the tactical and the operational activities by BIS is within the scope of the RDM for purchasing. This does not mean all relevant aspects can be found in the RDM. Where this is not the case, conclusions should be drawn with regard to on the one hand the application of data structures and on the other the support of today's purchasing software.

5110 Furthermore, the purchasing function encompasses preparatory activities that are separate from a specific purchase requirement. These are on a strategic and tactical level and concern decisions made by senior management as well as activities aimed at product, process and supplier development. Examples are:-

(a) policy regarding buying versus producing;
(b) choices for or against single sourcing;
(c) certification programmes for suppliers.

5111 Support for these activities by means of BIS should fall within the scope of the RDM. However, in the present generation of BIS little can be found in this area. Consequently these aspects (still) have not been dealt with in the RDM.

5112 The management of a purchasing department and the supervision of employees is considered to be non-purchasing specific and falls outside the RDM. Here, one should think of development, introduction and checking of guidelines, procedures, job descriptions, and the training of personnel.

Specification of purchasing groups

5113 Nearly every organisation has an enormous variety of goods and services to be purchased, in contrast with the sales of any given company where only a specific assortment is handled (the core-business). In the literature a distinction is often made in purchasing groups that are based on management differences. In terms of data structures an entirely different typology needs to be used, which is shown in figure 5.5.

5114 For the assessment of purchasing systems, however, it is necessary to pause by the customary grouping on the basis of control aspects. For following purchasing groups can be distinguished:-

(a) *raw materials* where next to the usual demand-driven situation (fulfilling a requirement) there can also be a supply-driven situation such as in the dairy industry;
(b) *additives* that are not physically part of the final product, but are used or consumed in the production process;
(c) *semi-manufactured products* usually these are simple products that have gone through several production stages and may be processed further at a later stage (such as sheet steel, rolled wire, alcohol). Possible characteristics include:-
    (i) mostly bulk goods;

(ii) limited assortment;
(iii) freely available on the open market;

(d) *components* with characteristics such as:-
  (i) standard versus specific components;
  (ii) quality aspect, preventing the production process from stagnating (quality is of course important everywhere, but especially when it concerns the more complex components);
  (iii) demands placed on the design or the specification;
  (iv) inspection, zero series, quality plans, and certification;

(e) *finished products* that are sold with relatively little added value (commercial environment). Specific characteristics include:-
  (i) purchasing on the basis of supplier specifications;
  (ii) a broad assortment;
  (iii) usually a direct relation between purchasing and sales;
  (iv) the commercial role that purchasing has as part of the trading activity;

(f) *investments or capital goods*, where the following is important:-
  (i) monitoring progress;
  (ii) quality;
  (iii) preliminary activities;

(g) *indirect goods or consumable goods* also known as MRO items (Maintenance, Repair and Operating Supplies). Characterised by:-
  (i) very large assortment;
  (ii) many unique products (for example spare parts often are tied to one company);
  (iii) low and irregular usage;
  (iv) collective invoices and annual agreements;

(h) *services* where we are concerned with subcontracting, contracting, consultancy, and leasing. Some interesting examples are:-
  (i) engaging temporary personnel;
  (ii) third party transportation;
  (iii) leasing capital goods;
  (iv) hiring consultants;
  (v) third party software development;
  (vi) negotiating insurance.

5115 There are some of purchasing groups that can perhaps theoretically be called purchasing, but where the information required and terminology used deviates to the extent that they are no longer considered as such. The most important examples are:-

(a) *raw materials futures,* not aimed at the acquisition but at taking options to cover risks;
(b) *periodical subscriptions,* where thought is only given to the value of a subscription when initially taken. The renewal of subscription becomes automatic;
(c) *employing personnel,* where labour terms could be compared to purchasing conditions;
(d) *telephone subscriptions,* which are unavoidable and cannot be influenced by a purchasing function;
(e) *consumption of products from public utilities*, where the same applies as for telephone subscriptions;
(f) *taxes,* an unavoidable legal obligation.

5116 Indeed, costs are incurred without actually purchasing anything. A practical condition of purchasing is that the costs must be influenceable. Naturally, it is possible to include these types of purchasing in the RDM. In practice, however, there is normally a need for support of a different kind than that offered by a purchasing system. A further elaboration is beyond the scope of this framework.

Typology of data elements and structures

5117 In paragraph 4251 a breakdown of data structures and entity types was given (see also figure 5.2) for inclusion in the scope of the RDM. Code tables, such as countries, foreign currencies, and payment conditions, should fall outside the scope of the RDM. There is a high risk that they would distort the understanding and add little value (known from experience).

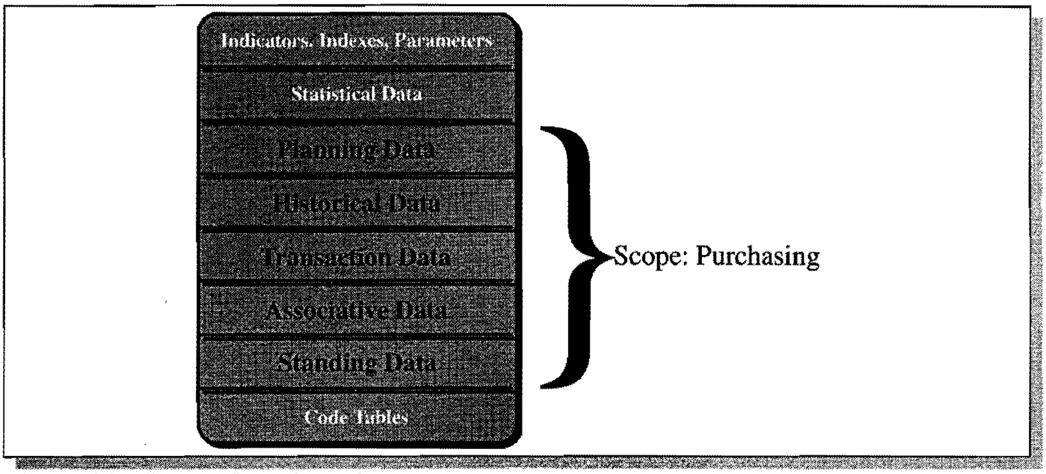


Figure 5.2 - Scope of purchase data

5118 The statistical data also fall outside the scope due to the fact that it is only composed of transaction and historical data, which is included in the assessment. This also applies to the indicators, indexes and standards. However, due to the importance within purchasing, explicit attention is paid to supplier performance.

5119 In the general conclusions regarding purchasing systems some variation on these boundaries is made.

Design of the RDM for Purchasing

5200 The full Reference Data Model (RDM) for purchasing is considerable and therefore difficult to assess for people other than the designer. To avoid this problem, the RDM is constructed in a number of steps, each step is described in detail. The following RDM component structures are discussed:-

(a) *supplier's organisation*, including the external organisation structure of the supplier, the actual creditor, and other relevant parts of the organisation;
(b) *typology of purchase items*, including a detailed structure of different kind of items and purchase items;
(c) *packaging material and purchase bills*, including alternative structures for dealing with (reusable) packaging and other accompanying delivered items;
(d) *purchasing conditions*, including standard conditions, quotations, specific conditions, and additional purchasing costs;

(e) *ordering*, the actual procurement by means of single orders, on the basis of blanket orders or fixed delivery schedules;

(f) *receipts*, including a structure that supports the (warehouse) receipt process and the settlement of returns;

(g) *requirements and destination*, monitors the status of various kinds of purchase requirements, including the details through to information about how these requirements are eventually met and the goods reach their destination;

(h) *invoicing*, including the different ways in which financial settlement can take place;

(i) *history*, including alternative approaches to maintaining a history of events;

(j) *alternative items and suppliers*, including variants used to determine alternative purchase items, suppliers, or combinations of the two;

(k) *delivery addresses*, including an elaboration of some special types of delivery to different receipt addresses;

(l) *multi-site structures*, including possible forms of multi-site support through purchasing systems;

(m) *controlling data*, including the remaining issues in the area of management and use of purchasing systems, such as the use of texts, change management, and supervision and task division among responsible employees within the purchasing department.

5201 For purposes of illustration, an existing BIS has been projected on the RDM.

Suppliers organisation

5202 Some supplier data has to be known in advance, i.e. before requesting a proposal or ordering any goods or services. Per '*Supplier*' the name and address data, the contacts, standard delivery, payment and discount conditions can be recorded. The supplier is defined here as the legal entity with which purchasing conditions are agreed. Sometimes more commercially relevant data is determined per supplier, like the size of turnover, number of employees, and market share.

5203 It can be useful to record the complex structure of some supplier organisation, for example if agreements are made with a central or co-ordinating body and orders (depending on the location of the requirement) are placed elsewhere. Figure 5.3 shows this structure in the RDM.

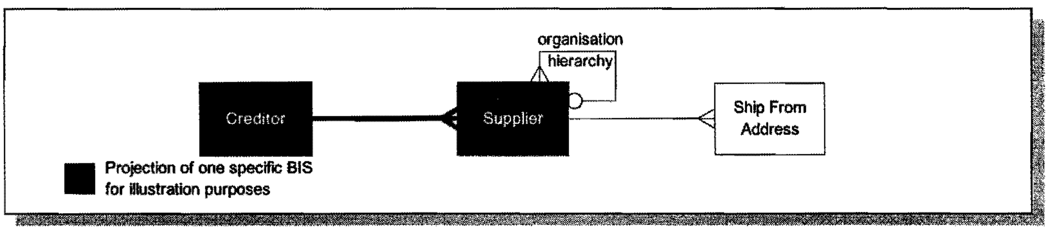


Figure 5.3 - Supplier organisation

5204 Because of the integration with the financial administration, each '*Supplier*' should also be known as a '*Creditor*'. Incoming invoices are preferably registered centrally in the accounts payable administration. If it concerns a purchase order it is passed on to the purchasing system (see also figure 5.21). The creditor is defined as the legal entity that sends the invoice. In integrated BIS the two data files '*Creditor*' and '*Supplier*' are often combined and each creditor can act as a supplier. In business co-operatives or in factoring where the financial settlement of deliveries is subcontracted in exchange for financial credit, the invoicing is done through legal entity other than the supplier of the goods or services.

5205 If transportation is arranged by the buyer (deliveries 'ex-works'), it is often necessary to distinguish multiple *'Ship From Addresses'* per supplier. This means that a supplier can have several warehouses at different locations from which goods can be supplied. Besides the address data, the transport times per *'Ship From Address'* should be recorded which can then be taken into account in determining the total delivery time. Without such an entity type this data has to be determined for individual purchase orders and altered, possibly making use of some default value per supplier.

Typology of purchase items

5206 While companies focus on a limited number of product/market combinations and supply a specific assortment on the sales side, the purchasing function is aimed at a great diversity of goods and services (ranging from raw materials and office supplies to capital goods, consultancy services, and leased cars). However, a distinction can be made in the way matters are treated and controlled. In terms of data structures, a categorisation based on distinctive relationship types and attribute types is useful.

5207 The starting assumption is that everything to be purchased should be uniquely identifiable. Therefore goods and services that can be purchased are defined as a *'Purchase Item'*. A purchase item therefore is an item or service for which purchase agreements can be recorded. Figure 5.4 shows this structure in the RDM.

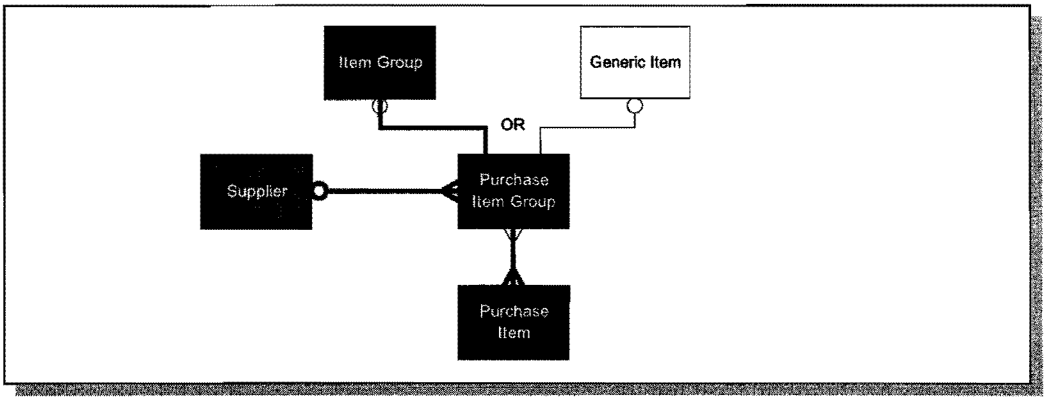


Figure 5.4 - Groups and individual purchase items

5208 Recording purchasing conditions can (still) take place at a more general level, for example per *'Purchase Item Group'* or for a *'Generic Purchase Item'* (generic items and product configurators have become quite common in the sales process, however purchase items can also be generically defined).

5209 Only when calling off or ordering, are the complete specifications necessary (for example: the price agreement is for wood per m^3, only when actually ordering should the required measurements and quality be recorded). In various places in the RDM this distinction between *'Purchase Item Groups'* and specific *'Purchase Items'* is made. If necessary, different groupings of items can be made, so that a specific purchase item can belong to different groups. Useful examples could relate to various supplier groupings such as for price agreements and for purchase planning. This makes demands on the BIS functionality as it has to be capable of handling such combinations. This complex relationship will not be found in most BIS.

5210 As is shown in figure 5.5 the entity type '*Purchase items*' is further specialised. The first specialisation of '*Purchase items*' is that of '*Goods*' versus '*Services*'.

Goods

Assortment items

5211 Assortment items are standard items in the context of one particular company, and so this does not depend on what customers or suppliers consider to be standard. For these items a series of so-called standing data is recorded, for example regarding the control of stock levels, cost prices, units and usage.

Generic items and variant items

5212 Generic items are not individually specified and registered within the information system. These items have characteristics that can be described as collections of possible features (domain), and in addition the exclusions of combinations of features not permitted. Defining these features leads to the generation of a specific variant item. If necessary, variants can be kept in stock in contrast with generic items. For the specification of generic item variants a product configurator is used. However, these are still mostly known and applied in the sales process.

Special items

5213 Special items are basically complex and project-related for which no standard item data is recorded in the information system. Extensive specification takes place at the time the (purchasing) requirement arises. Every time a request is made for a special item, the specification will differ. It is therefore impossible to define and specify these items in advance. As a result, these items are indeed coded but, for example, on the basis of the purchase order number, and not based on the standard data in an item master file. There is no additional work in maintaining the data such as authorisation, non-relevant data fields, and the removal of unused items.

5214 The coding of these items to trace the (administrative and logistics) processing has to take place in a different way, i.e. by numbering these items contiguously or by using the purchase order line as an identification. In practice purchasing these items is sometimes referred to as subcontracting. This is a different kind of subcontracting (namely without the supply of material) than is referred to later in the RDM, see the definition in paragraph 5221 of subcontracting as a service. If the high requirements of the initial process (requirements and quotation) and order monitoring are not necessary, then '*Cost Items*' can be an alternative to this '*Special Item*'. This means in fact that most management is kept outside the information system.

Cost items

5215 Cost items are understood to be (usually standard) items of which the purchase value (per piece, per order, or per annum) is so low that no standard purchasing data is stored. Typical examples are the smaller office supplies such as pens and pencils.

5216 As soon as the purchase value crosses a critical line and the organisation starts to (cost) effectively concern itself with the purchasing of these items, they automatically become '*Assortment Items*'. The purchasing of cost items only takes place at the level of an '*Item Group*', where a textual specification is recorded in the purchase order.

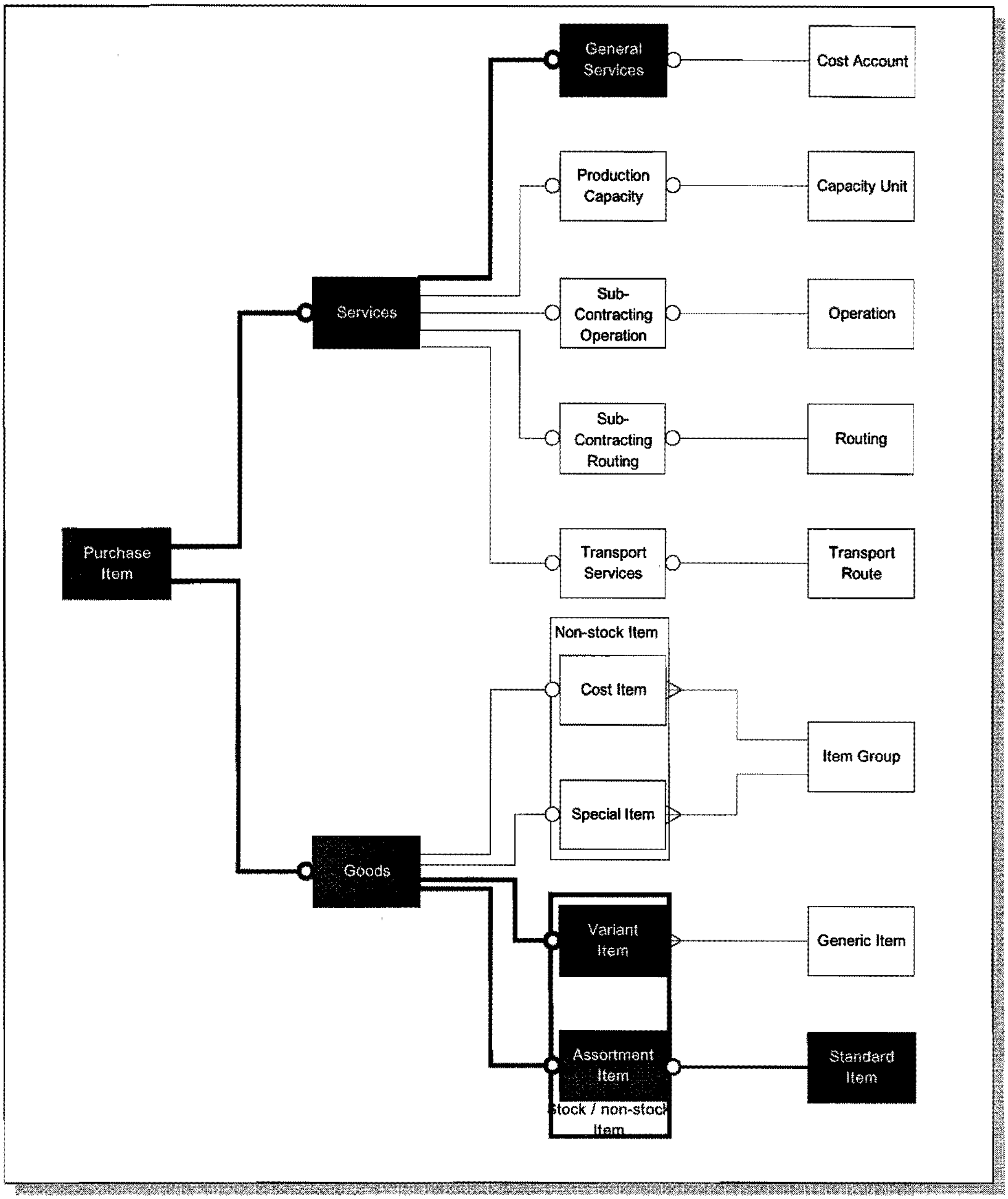


Figure 5.5 - Typology of purchase items

Stock items and non-stock items

5217 The '*Goods*' described can be stock-controlled. According to the definition used earlier, these are '*Assortment Items*' or '*Variant Items*'. In the information system a stock should never be registered for '*Cost Items*' because of their low value, see paragraph 5215. The '*Special Items*' are unique by definition, so that these in any case, will not be controlled as (anonymous) stock.

5218 Thus stock might only be recorded for the '*Assortment Items*' and '*Variant Items*' within an information system. Consequently control and reporting is possible, for example: a replenishment mechanism and cycle counting. This in contrast to goods that may be stored in a warehouse, but are not controlled as stock items by the information system. These can be working stocks that are controlled by a kanban-system or by defaults, and special project-related objects that are controlled using the project planning

Services

Transport services

5219 In this context transport services are all activities concerning the transportation of goods between storage or processing locations, suppliers, customers and locations under company management. Transport services are executed by third parties but under instruction from the purchasing function. Transports are defined in terms of routes (to and from location), the transport means and possibly the nature of the load. In comparison to other purchase items, one is concerned with price agreements per route, or tonnage per kilometre, and per fully loaded truck.

5220 If management considers transportation to be of secondary point of importance (due to the small share of the costs, little variation in carriers and routes, and simple agreements and conditions), then more general solutions could be a good alternative. One solution is to consider transport as a '*General Service*' (see paragraph 5226), or by defining the different routes as '*Goods Items*'; note that the completed transportation then has to be received just as for normal items.

Subcontracting routing

5221 Subcontracting implies production activities carried out externally. One of the characteristics is that material can be supplied under the responsibility of the subcontracting organisation.

5222 One of the forms that can be distinguished is the subcontracting of complete production orders. This form is already determined on the item level in the form of a completely externally treated '*Recipe*' or '*Routing*' (see also chapter 6). The material to be supplied is specified in the '*Bill of Material*'. This is also called subcontracting from the warehouse.

Subcontracting operation

5223 Another type of subcontracting that can be distinguished is the subcontracting from production. Here, these are individual operations that are carried out externally, alongside the internally performed operations on the product. In contrast with the previous form, these are usually simple standardised operations like galvanising. These '*Subcontracting Operations*' are defined in '*Recipes*' or '*Routings*' just like the normal '*Operations*'.

5224 If these types of subcontracting are not supported, the alternatives are:-

(a) treating them as the purchase of normal items, and solving the material supply and production management aspects procedurally;

(b) treating them like normal operations or production orders, and taking on the purchasing aspects such as invoice checking procedurally.

Production capacity

5225 This refers to the hiring of '*Production Capacity*', which is used in the production process. The types of capacities are defined in the '*Capacity Units*', in the same way one's own production capacity is arranged (these are recorded in the recipes and routings, see also chapter 6).

General services

5226 The '*General Services*' concern the various purchases that are not related to the primary business process (such as utilities and investments) and therefore are not further detailed in this context. These purchases are related to '*Cost Accounts*' and '*Cost Centres*'.

Packaging material and purchase bills

5227 The packaging material in which deliveries are made may be recorded and controlled for several reasons:-

(a) its influence on the purchasing conditions, such as the price or transport costs;
(b) the specific demands that are placed on the packaging material;
(c) reusable packaging (as with receipts, see also figure 5.17 where further details are given);
(d) a certain deposit value;
(e) stock-keeping due to management considerations or the value;
(f) the management of 'pool' packages (standard packaging material that is used by several companies, for example Euro pallets).

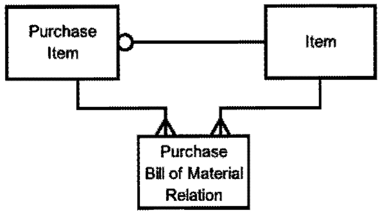
5228 Most packaging material is not purchased but returned to the supplier. The mechanism to control this is based on a deposit or is without financial consequences (except in the case of loss). Another variation is the time at which empty packaging material is returned, i.e. directly after having received the goods through exchange (this is by far the easiest way, as there is no registration or financial transaction necessary) or when the packaging material is emptied. In the latter case a more extensive administration is necessary.

5229 See later paragraphs for more details on agreements and purchasing conditions. Furthermore, the purchasing function often has a special responsibility for registration and monitoring.

5230 Sometimes there is a hierarchy of packaging materials: box, outer box, pallet/container, and other packaging material. The above aspects will apply to some of these materials, but not to others. Figure 5.6 gives three possible implementations in the RDM, by means of:-

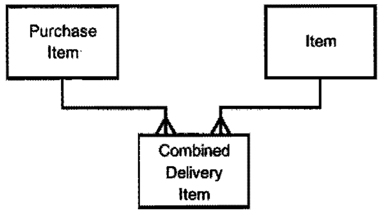
(a) a '*Purchase Bill of Material*';
(b) an entity type '*Combined Delivery Items*';
(c) an extension of the entity type '*Purchase Item*'.

5231 The '*Purchase Bill of Material*' gives a general data structure that is not only applicable to packaging material. Packaging material is one of the possible specialisations of '*Item*'. '*Purchase bills of material*' can also be used for the management / control of the components used in the items ordered from the supplier, for example a type indication of microprocessors used in purchased PC's to make it possible to replace them yourself. besides this, '*Purchase Bills of Material*' can be used to determine the materials to be supplied in subcontracted production orders (see figure 5.5, '*Subcontracting Routing*').



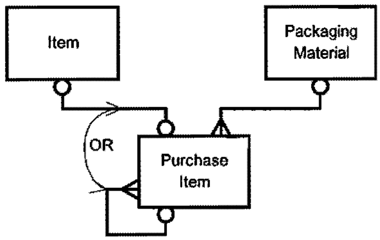
Purchase Bill-of-Material

Purchase Item	Item	Quantity
palette paint	palette	1 piece
palette paint	box paint	45 pieces
box paint	box	1 pieces
box paint	can paint	24 pieces
can paint	can	1 piece
can paint	paint	1 litre



Combined Delivered Items

Purchase Item	Combined Delivery Item	Quantity
palette paint	palette	1 piece
palette paint	box	45 pieces
palette paint	can	45 x 24 pieces
palette paint	paint	45 x 24 litre



Purchase Item

Purchase Item	Main Purchase Item	Item	Packaging Material	Quantity
can paint	-	paint	can	1 litre
box paint	can paint	-	box	24 pieces
palette paint	box paint	-	palette	45 pieces

Figure 5.6 - Packaging material and purchase bills

5232 A more simple data structure makes use of an entity type called '*Combined Delivery Items*'. For a specific '*Item*' (or '*Item Group*') the packaging material to be used is defined. One limitation is the lack of a hierarchical structure of packaging materials. However, the actual data structure is identical (up to the attribute level) to the implementation by means of a '*Purchase Bill of Material*'. The distinction between both solutions only exists at a functional level (the hierarchical / recursive interpretation by the software).

5233 One explicit implementation is the extension of the entity type '*Purchase Item*' with packaging material as an attribute type. To implement the hierarchical packaging structure, an attribute type called '*main purchase item*' has to be added too. The functional possibilities of this solution for the treatment of packaging material is equivalent to that of '*Purchase Bills of Material*'. It should be noted that the '*Purchase Item*' here is no longer a specialisation of an '*Item*'. For example, this could mean that a stock administration, which is normally standard for '*Items*', might not be possible or is only partially supported for '*Purchase Items*'.

5234 Figure 5.7 shows the combined RDM. Here, it is striking that:-

(a) the implementations '*Purchase Bill of Material*' and the extension of '*Purchase Item*' are suitable for the support of a hierarchical structure;

(b) the solutions '*Purchase Bill of Material*' and '*Combined Delivery Items*' can also be suitable for other purposes than the packaging material administration;

(c) here the hierarchical extension of '*Purchase Items*' exists, attention should be paid to the presence of support for the desired stock control function..

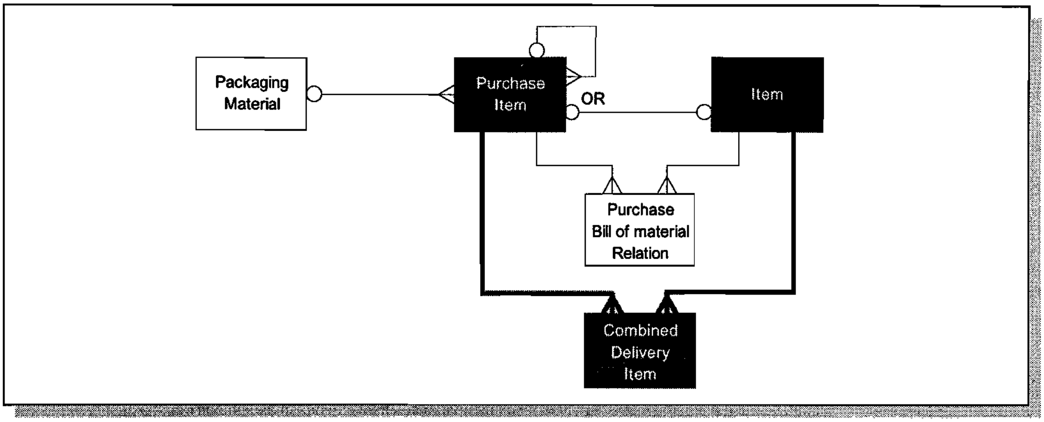


Figure 5.7 - Combined structure for packaging material and purchase bills

5235 A real '*Purchase Bill of Material*' seems to offer the most combinations of options.

Purchasing conditions

5236 Purchasing conditions are defined here as all conditions under which purchasing takes place without a financial obligation. (these are recorded in the actual order, see figure 5.14). The purchasing conditions mainly concern the following:-
(a) quality;
(b) price (per unit);
(c) discount;
(d) delivery time;
(e) delivery conditions;
(f) additional costs;
(g) payment conditions;
(h) service conditions;
(i) ownership conditions (types of leasing / holding in consignment / rental);
(j) conditions and obligations for returns;
(k) information about the products to be delivered (documents and certificates);
(l) agreements about transportation.

5237 The purchasing conditions can be recorded both dependently and independently of the formal obligation. Distinction can be made between three types of purchasing conditions (see figure 5.8):-
(a) *specific purchasing conditions* (negotiated by the purchasing company), divided into:-
  (i) *singular purchasing conditions*, that are usually dependent on the (size of a) possible order and are therefore directly related to a purchase order or blanket order;
  (ii) *continuous purchasing condition,* that are usually independent of any possible order;
(b) *standard purchasing conditions*, that are fixed by the supplier and upon which the purchasing company has no influence. These are normally contained in the standard price list.

5238 Specific purchasing conditions may be the basis of quotations and quotation negotiations. Here the quotations are dealt with before the specific purchasing conditions.

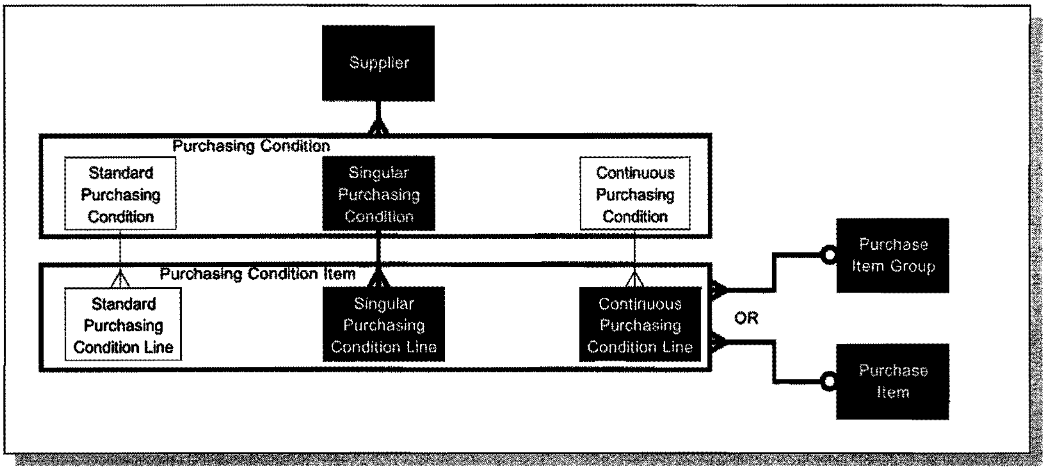


Figure 5.8 - Purchasing conditions

Standard purchasing conditions

5239 The standard purchasing conditions are usually called 'price lists'. Many suppliers have fixed prices (and sales / delivery conditions) for standard products (or services). These conditions are fixed by the supplier and the purchasing company have no negotiating influence. This, in contrast with the specific purchasing conditions and agreements, negotiated between purchaser and supplier that will be dealt with later.

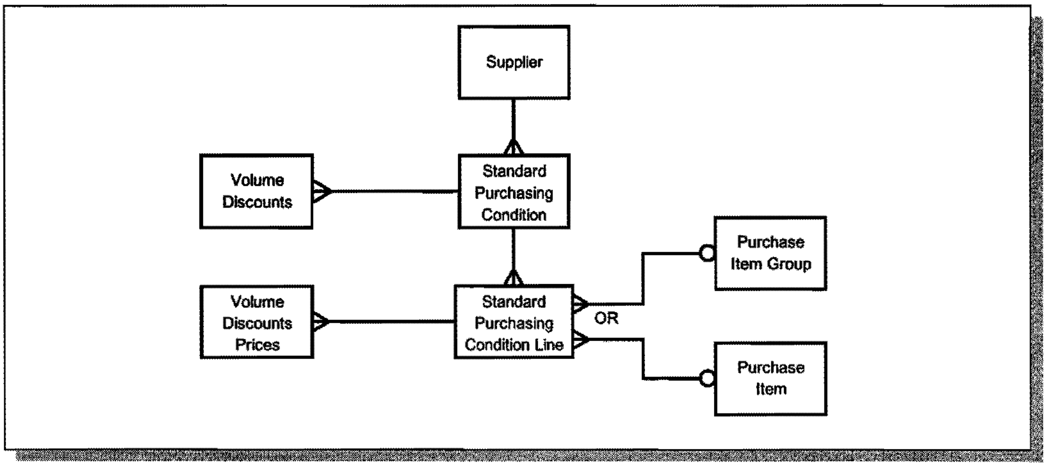


Figure 5.9 - Standard purchasing conditions

5240 *Standard Purchasing Condition Line*: Per '*Purchase Item*' standard conditions are recorded as mentioned earlier. Additionally, the supplier's identification of the purchase item is recorded here (for example supplier's item code). Sometimes standard prices, volume prices or discounts (quantity rebate) are also recorded here.

5241 *Standard Purchasing Condition*: Because of the limited validity (from and to dates) it is useful to have several versions of the purchasing conditions. At this highest level an overall discount percentage or volume discount can be relevant and thus the validity should be given by means of 'from' and 'to' dates.

5242 For the most important purchasing groups, the supplier's standard prices and conditions are not good enough. Here, the commercial role of the purchasing function begins and one of the aims is gaining the most suitable conditions for the organisation, both in financial and logistics terms.

5243 The request for a quotation put to several suppliers supports this role. In simple situations this can requested informally (for example by telephone), but for more important items it can be necessary to specify and record this request for a quotation. Such a request may consist of several '*Purchase Items*' (or '*Purchase Item Group'*) usually including extensive textual information. The request is then forwarded to a number of potential suppliers.

5244 If the supplier reacts with a concrete offer, this will be recorded as a purchase quotation (and quotation lines with the specification of purchase items). Thus, it can occur that quotations are recorded without a formal request for quotation having taken place.

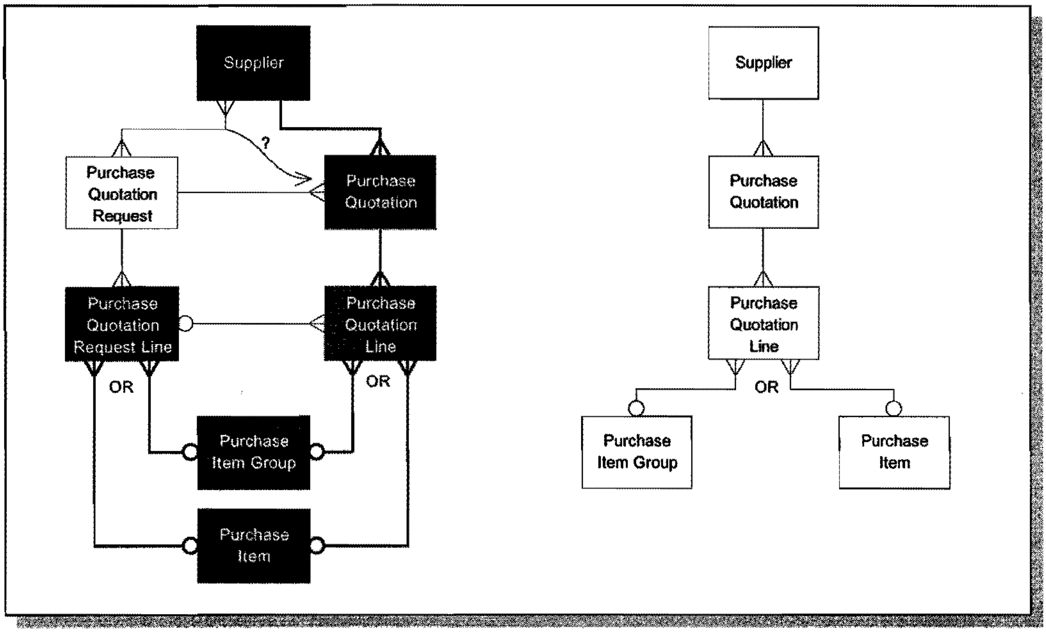


Figure 5.10 - Purchase quotations

5245 The n:m relationship between '*Supplier*' and '*Purchase Quotation Request*' on the left in figure 5.10 cannot simply be implemented and first requires normalisation. This leads to a new entity type between '*Supplier*' and '*Purchase Quotation Request*' with exactly the same relationships as the existing entity type '*Purchase Quotation*'. Though the meaning of these relationships is not exactly equal, various BIS do not make a distinction between the two. The consequence is that one is forced to conclude which relationships are being referred to based on the value of the attribute types.

5246 A simplification that has been chosen in many implementations is the combination of the entity types '*Purchase Quotation Request*' and '*Purchase Quotation*'. Some redundancy is introduced with this, especially if identical requests are sent to many suppliers. However, modern purchasing practice teaches that requesting quotations from as many suppliers as possible is not desirable. It is better to make a good selection beforehand and to devote time and energy to a limited number of suppliers. From this perspective the combination of '*Purchase Quotation Request*' and '*Purchase Quotation*' into one entity type might be acceptable. However, the consequence is that when a supplier makes several quotations during negotiations, only the last quotation can be retained.

5247 These simplifications lead to the structure on the right-hand side in figure 5.10. Though both data structures can be found in purchasing systems, the right-hand one appears to have a number of limitations.

Specific purchasing conditions

5248 Accepting a quotation leads to conditions that apply to a specific situation. In the established quotation, conditions such as prices, discounts, and delivery times are recorded. Accepting a quotation means that these conditions have been defined in separate attribute types and have been explicitly recorded as purchasing conditions. Here, a distinction has to be made between singular and continuous conditions and agreements. '*Singular Purchasing Conditions*' apply to one '*Purchase Order*' or '*Blanket Order*' (see figure 5.14) and are usually directly stored in these entity types. Though the '*Singular Purchasing Conditions*' are not recognisable as separate entity types in most BIS, they have been normalised in this RDM for the sake of clarity. Figure 5.11 shows the accompanying data structure.

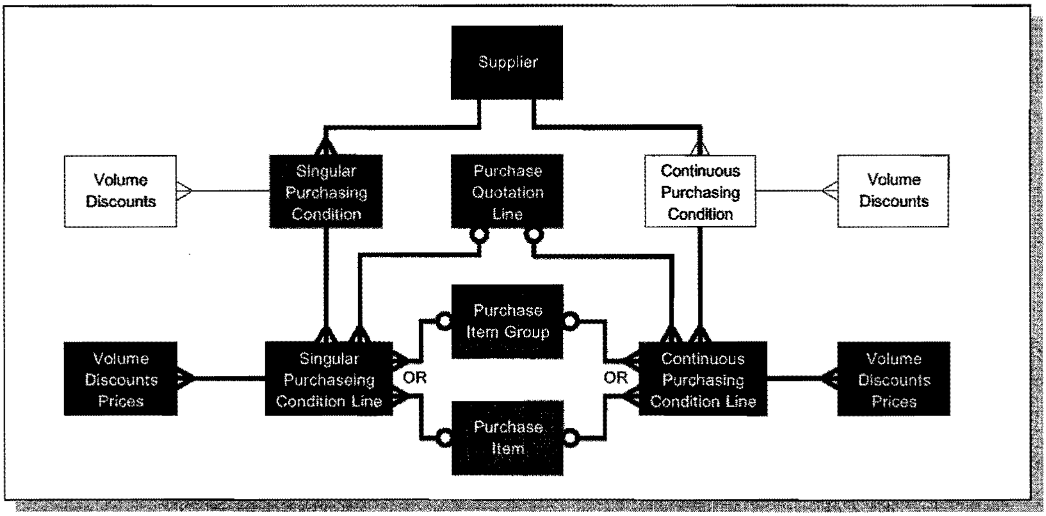


Figure 5.11 - Specific purchasing conditions

5249 Normally order quantities are fixed in these '*Singular Purchasing Conditions*'. However if this is not the case, it can be useful to work with volume prices and discounts, like in the '*Standard Purchasing Conditions*'.

5250 '*Continuous Purchasing Conditions*' have a certain period of validity. During this period several purchase orders or even blanket orders can be closed against these conditions. As with the '*Standard Purchasing Conditions*', volume prices and discounts per '*Purchase Item*' can also occur.

5251 In implementations these *'Continuous Purchasing Conditions'* are nearly always combined with the *'Standard Purchasing Conditions'* (see figure 5.9) and a distinction is not visible in the data structure. This distinction can be of importance, for example, in establishing the performance of the purchasing function, since the *'Continuous Purchasing Conditions'* are the result of negotiation.

Additional purchasing costs

5252 Possible additional purchasing costs can be part of the agreed purchasing conditions. Examples are packaging costs, dispatchment or transportation costs, costs for the production of documents, and other costs often related to third parties. These costs may be related to the whole *'Purchase Order'*, but can also relate to one or more specific *'Purchase Order Lines'*. When describing receipt (see figure 5.17) the additional costs related to shipments and receipts will be discussed.

5253 Recording *'Additional Purchasing Costs'* explicitly as a separate entity type gives the opportunity to record multiple costs, each with a cost type and background for calculation and possibly a different creditor (for example document costs might be charged by a customs shipping agent).

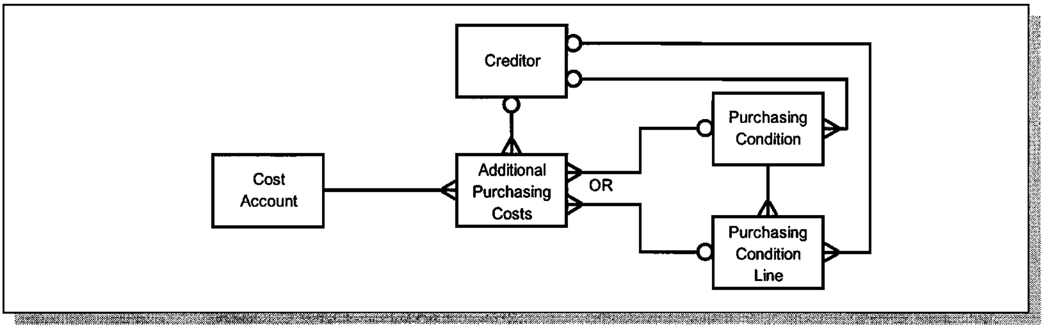


Figure 5.12 - Additional purchasing costs

5254 Many systems have these costs stored as a fixed part of the *'Purchasing Conditions'* and therefore have limited them to a few predetermined types. The lack of a separate entity type *'Additional Purchasing Costs'* is no problem for most purchasing situations and can be preferred in simple implementations (no extra screens, simpler input, easier recall).

Ordering

5255 Procurement is one the most important operational activities of purchasing. Within which three types can be distinguished, see figure 5.13:-

(a) *the standard purchase order*, traditionally the most common type;
(b) *the blanket order*, where call-off can be made on;
(c) *the delivery schedule*, an order form that is based on predefined time-phased requirements for goods or services (deliveries are made conform the schedule).

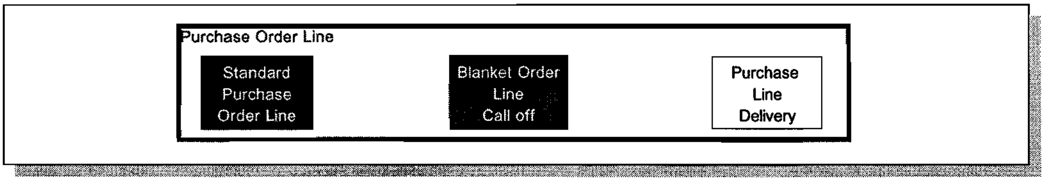


Figure 5.13 - Typology of purchase orders

Standard purchase orders

5256 The oldest type of order is the purchase order or single order. Each time a requirement for a purchase item arises, a purchase order is made using the current or negotiated purchasing conditions.

5257 Per '*Purchase Order Line*' the specific purchase item is identified. Sometimes the '*Batch*' (or lot number) is known when ordering or is specifically ordered. For example if batches have been inspected beforehand by the purchasing company or if exactly the same characteristics are required as in a previous delivery, for example the same batch colour.

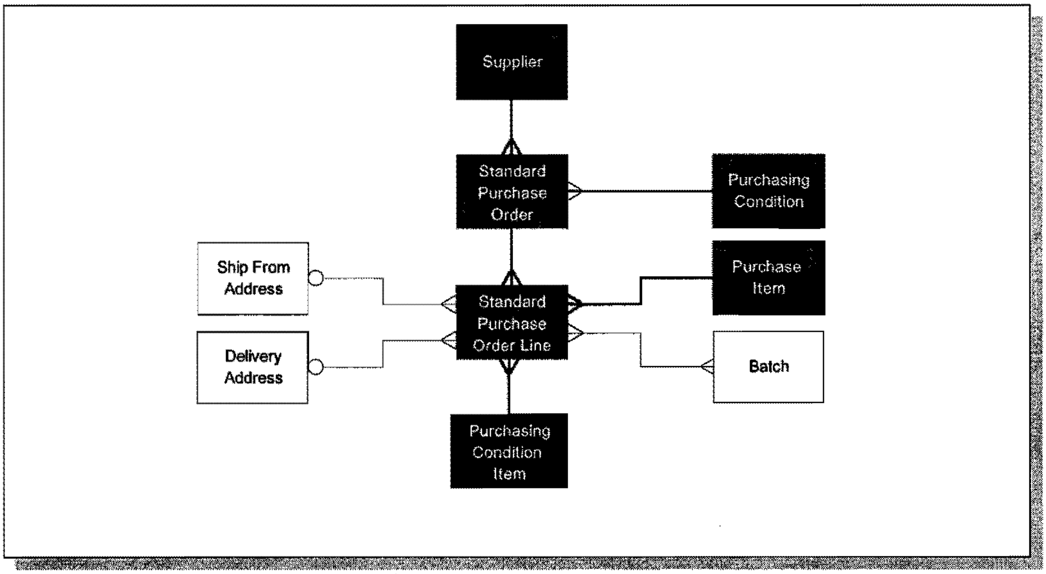


Figure 5.14 - Standard purchase order

5258 If transportation is carried out under one's own management, the '*Ship From Address*' has also to be determined too. The '*Delivery Address*' for goods or services rendered, must be known at all times.

Blanket order

5259 There are a large number of variants and types of contracts. The '*Blanket Order*' is used here as a collective name for all of these types. A blanket order is defined as an agreement about price, quantity to be purchased within a given period and various conditions, such as under-delivery and over-delivery. The quantity must be called off within the specified period. The situation in which a delivery schedule is known in advance will be discussed later.

5260 In practice many types of contracts exist with different names. There are no clear guidelines for names. All these different types of contracts and agreements are related to the following:-

(a) *the total quantity* to be purchased (expressed in money, volume, or in some other unit). Besides a fixed contractual quantity, agreements can be made about the maximum quantity variations under the same conditions. A special variant is the reservation of capacity at the supplier within which call-off orders can be made for items to be produced;

(b) *the period* in which the contractual quantity has to be purchased. Besides flexibility in this period, agreements can also be made for an unspecified period (standing order or open-end contract). For these variants the contractual quantity should be interpreted as the quantity per period;

(c) *the delivery time.* Besides a fixed delivery time per call-off, agreements can be made about shorter delivery times if the supplier is informed beforehand about the required quantities (with a degree of certainty, for example allowing the supplier insight in the planing);

(d) *the call-off quantity.* Fixed quantities that are ordered for or called off at certain moments in time to be determined later. The call-off quantities may be variable, with lower and upper limits;

(e) *the call-off frequency.* On fixed dates a quantity to be determined at a later is called off. In the contract negotiations specific (singular) purchasing conditions are usually agreed;

(f) *the quality requirements.* Besides textual description quality is usually concerned with the tolerance of measurements.

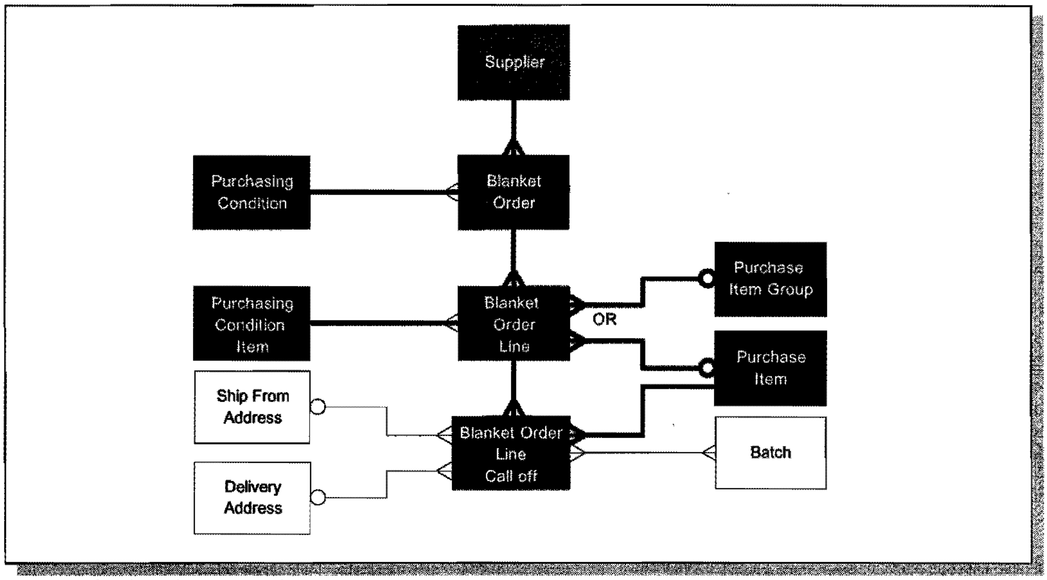


Figure 5.15 - Blanket order

5261 Where the contract agreement is still related to a '*Purchase Item Group*', a specific '*Purchase Item*' has to be fully specified at the time of ordering.

5262 Just as with the '*Singular Purchase Order*' the '*Batch*' can be important in certain cases (see earlier). This also applies to the '*Ship From Address*' and the '*Delivery Address*' for goods or services, which must be known by the time a call-off order is requested.

Delivery schedule

5263 While the call-offs on blanket orders are made individually at a later date, '*Delivery Schedules*' are known beforehand. Delivery schedules can be seen as a further detailing of the '*Purchase Order Lines*' agreeing delivery moments. The delivery schedule is useful if:-

(a) a time-phased requirement is available beforehand;

(b) there is a kind of 'subscription' for the automatic delivery of an agreed quantity;

(c) a 'rolling forecast' is implemented under the condition that the first few deliveries have been fixed (firmed orders). A status indicator for the deliveries that follow, is used as an indication of the required quantity for the next deliveries. An alternative implementation of the rolling forecast is possible by using time-phased purchase requirements instead of purchase orders.

5264 *'Delivery Schedules'* are particularly useful for reasonably predictable requirements and they offer an opportunity to limit costs of holding stocks. Through such (good) agreements a closer tie grows between the suppliers and their customers. Nevertheless some kind of feedback mechanism can be important, and for real 'rolling forecasts' this is essential.

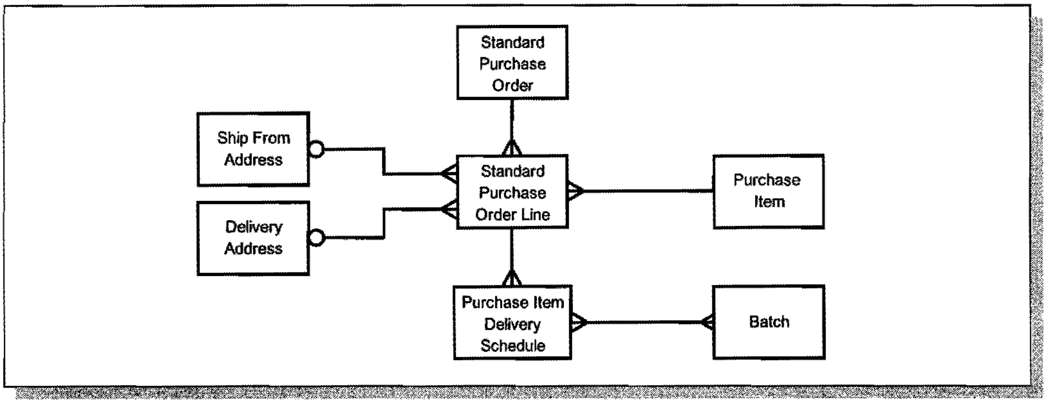


Figure 5.16 - Delivery schedules

5265 An alternative solution that can be found in some systems uses several purchase order lines for the same item but with different delivery dates. Of course, entering such a *'Delivery Schedule'* manually is possible in nearly every system as long as the delivery date is in the order line (and not in the order header). Some BIS support this form of delivery schedules by automatically copying this redundant data.

5266 A disadvantage of this method of implementing delivery schedules are alterations mostly effecting all order lines. Let us assume that the purchasing conditions change or that a textual explanation has to be altered, then this will have to be changed separately for each purchase order line due to the redundancy in the data structure. Software constraints make it almost impossible to do this automatically and changes will therefore have to done manually (per order line).

5267 Nevertheless for occasional use this simple solution is easier than the normalised implementation given in figure 5.16. For advanced purchasing situations where 'rolling forecasts' are used or there is a predictable time-phased requirement, such an explicit implementation (the normalised version) of *'Delivery Schedules'* is worth while.

5268 There is no fundamental difference between the data structure of *'Delivery Schedules'* and call-offs on *'Blanket Orders'*; it only seems to be a small difference in terms. As there are many functional differences between the two in application, both structures are shown in the RDM. When assessing the data structure of an arbitrary BIS this means that care has to be taken when drawing conclusions based on the data structure alone.

Receipts

5269 Purchase orders or call-off orders of goods eventually lead to a receipt(line). Partial receipts, multiple receipts from one purchase order line, are often possible too. Receiving (and therefore the registration of receipts) is typical for purchased goods and not for services. However, most BIS do not make this distinction.

5270 A receipt concerns one shipment that is delivered by the carrier at a given moment. A receipt consists of several lines, where the quantity and quality of each item is determined and recorded. When immediate inspection takes place the relevant data (in attribute types) is recorded per receipt line. Other ways of recording inspection data are:-

(a) a separate status (and additional attribute types) that are added to a '*Batch'*;

(b) a registration of goods received in a separate warehouse or location;

(c) a separate entity type '*Inspection*' with a relationship to '*Receipt Line*'.

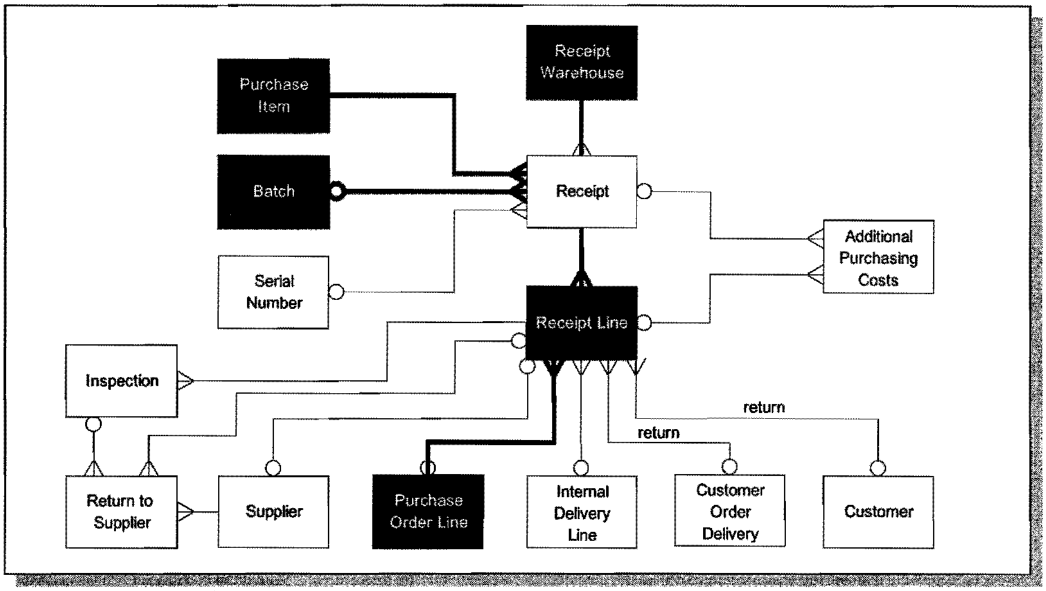


Figure 5.17 - Goods receipts

5271 The possibility to record unplanned receipts implies that a relationship is lacking with the purchase order line (which therefore is optional). If the goods are accepted, for example the a purchase order was made by telephone, this means there only is a separate relationship type with the '*Supplier*'. This option is sometimes used in the second case to establish the relationship between the receipt line and the purchase order line.

5272 The completion of returns to the supplier can be implemented in different ways. Depending on its importance the following data structures are to be preferred (listed in order of decreasing impact on the data structure):-

(a) a separate entity type for recording '*Returns to Supplier';*

(b) a separate attribute type in '*Receipt Lines*'. If the original purchase order no longer exists or can be traced, because the goods are returned much later, this must be solved as an unplanned receipt (of zero units) on which the goods can be returned;

(c) as a negative receipt. Consequently there is no direct relationship present with the originally registered receipt, other than via the purchase order in question (if it still exists in the information system).

Tracing returned packaging material is only possible in the first two solutions. If redelivery of returns is required after rejection during quality inspection, this can be realised in all three solutions, by adding this to attributes of the specific entity types.

5273 The function 'goods receipts' with the accompanying administration, the receipt inspection etcetera, is not specific for *purchased goods*. This structure should also apply to the receipt of returned goods from the *sales process* and the receipt of goods that have been transported from another site or warehouse, therefore the relationships with these other entity types are shown here.

5274 In the receipt process certain costs can become visible, for example on accompanying documents (like transportation costs). Recording these costs is useful for invoice checking later. However, the fact that these costs have a direct relationship with the receipt does not mean that they can always be recorded at that time.

5275 This form of receipt registration is based on a more traditional type of purchasing, in which each receipt is compared with the purchase order, the inspection takes place, the goods are stored in the warehouse, and finally the supplier sends an invoice. With the more advanced types of purchasing the activities of administration are reduced to an absolute minimum on the basis of good agreements. In its most extreme (most optimal) form this means:-

(a) no registration of receipts and therefore no verification with the order or the call-off order;
(b) direct receipt at the place of use (on the shop floor or by means of cross docking immediately to expedition);
(c) no registration of the called off quantities with respect to the contractual agreements. This registration takes place only at the supplier instead of at the customer as well as the supplier;
(d) quality control is not necessary, considering the supplier's quality guarantees;
(e) no goods invoicing, but automatic payment following usage ('pay on receipt' or 'pay on production').

5276 In the data structure this is especially characterised by the absence of '*Receipts*' and '*Purchase Invoices*'. Instead, for example, in production should lead to the automatic generation of payment suggestions. If necessary the information system can automatically make '*Purchase Invoices*' that directly have the status 'checked'. Such short-cuts in registration activities are visible in the data structure to a very limited extent.

Requirements and destination

5277 The purchasing function is always a part of a larger organisation. Purchasing does not function without someone or something requiring material, a service, or any '*Purchase Item*'. In this requirement coverage three stages can be distinguished:-

(a) the initial requirement;
(b) the allocation or coverage of this need (making sure that the requirement can be satisfied);
(c) the transaction or realisation (the actual provision in this requirement).

These three levels are discussed below.

Requirements

5278 The requirement is defined as the demand for items (goods or services) at a given time and quantity. Attribute types include the item (possibly a textual specification only), the quantity, and the date.

5279 Requirements can arise in different places within a production organisation as a result of:-

(a) sales on the basis of a (planned or actual) customer request;
(b) the replenishment of stocks in the warehouse;
(c) production, where the necessary raw materials, components or other items are needed, and also for a subcontracted operation;
(d) planning, based on a demand forecast from for example an MRP calculation;
(e) any other department where all kinds of (purchase) items may be needed.

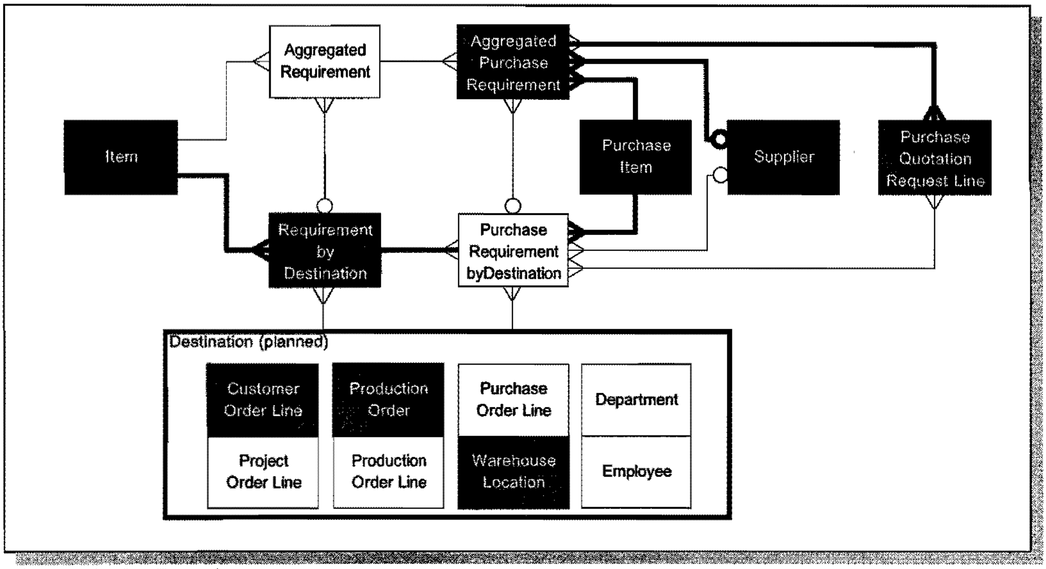


Figure 5.18 - Requirements

5280 The '*Destination*' is the entity type that generalises all these places in the organisation. The '*Destination*' is therefore defined as the one who has placed a requirement on an '*Item*' and for whom these '*Items*' are eventually intended. In terms of entity types these can be:-

(a) *the customer order line*. Analogue to the definition used in purchasing these can also be call-off customer orders on blanket customer orders or customer delivery schedules;
(b) *the production order*, where subcontracting is important (for an elaboration see paragraph 5400 and following);
(c) *the production order line*;
(d) *the project order line*;
(e) *the transport order line*, as a specialised type of a purchase order, that in fact subcontracts transport (see also paragraph 5400 and following);
(f) *the warehouse and location,* also when a requirement arises from an MRP calculation the actual destination is a decoupled stock point and therefore a warehouse;
(g) *the department,* which is usually a cost centre where this independent requirement arises;
(h) *the employee,* who signals an independent requirement.

5281 The requirements per destination can be aggregated for all destinations. A fundamental choice made in the logistics controlling structure (and also in the system) is whether the purchase decision, monitoring, and allocation takes place on the basis of this total aggregated requirement, or on the basis of each individual requirement (per destination). In a specific situation of course only one of the two must be chosen.

5282 In the latter case each individual requirement is translated to a specific '*Purchase Order Line*' and in turn which '*Receipt Line*' trace back to it. During the whole purchasing process all reservations can be followed in detail. Barring interventions, receipts will automatically have their original destinations.

5283 If control only occurs on totals (aggregated levels), then again a choice has to be made after receiving the goods as to how each '*Receipt Line*' is allocated (to which destination). If part of an order is delayed it can not be deduced which destination will suffer; the allocation decision is explicitly delayed.

Purchase Requirements

5284 The purchase requirement is that part of the requirement that is to be satisfied by purchasing (for example the rest might be satisfied by own production or is available from stock). The logistics controlling structure defines to what extent a requirements is always fulfilled via purchasing. Sometimes alternative ways to satisfy such requirements can be chosen, for example in-house production or supplying from different storage locations.

5285 Similar to the purchase requirements, comparable entity types can be defined for production (the production requirements) or supply from stocks (generates a replenishment requirement). Within the purchasing RDM these variants are not discussed. Again the purchase requirement can be divided into a '*Purchase Requirement per Destination*' and an '*Aggregated Purchase Requirement*'.

5286 Often it is known at the moment a given (item-dependent) requirement arises that it has to be purchased. In more complex situations, the production or purchasing of the same item could be an option. Also internal delivery from a different warehouse sometimes is an alternative. Companies where these kinds of choices are possible benefit from a division of the '*Requirement*' and the '*Purchase Requirement*' into different entity types.

Purchase coverage or allocation

5287 A purchase requirement leads eventually to the placement of a purchase order (line), making a call-off on a blanket order, or the coverage of this purchase requirement with already existing (independent) purchasing activities. Naturally, these expected goods (or other deliveries of services) should not automatically be made available to any other than the original destination (where planning does not occur at an aggregated level, see the choice listed under the purchase requirement in paragraph 5281).

5288 The purchase coverage or allocation is the purchase order (or other expected delivery) that covers the purchase requirement. At this level it may already be useful to determine batch numbers. Characteristic attribute types for this coverage are: item, quantity, date, batch number, purchase order and destination. Here, too, we can distinguish '*Purchase Coverage for Destination*' and '*Aggregated Purchase Coverage*'.

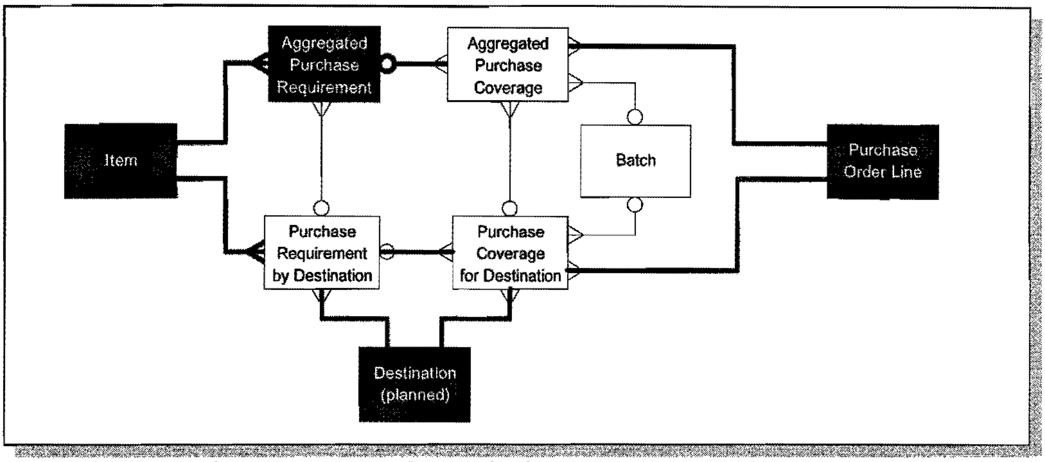


Figure 5.19 - Requirement coverage by purchasing

Purchasing transaction

5289 After receipt, goods must go to their final destination. Thus, a received batch can be split up and sent to several destinations. The determination of this destination can be made or traced in the *'Purchase Coverage'* and the *'Purchase Requirement'*. The *'Purchasing Transaction'* therefore represents the physical delivery of items from the purchase receipt line to their destination.

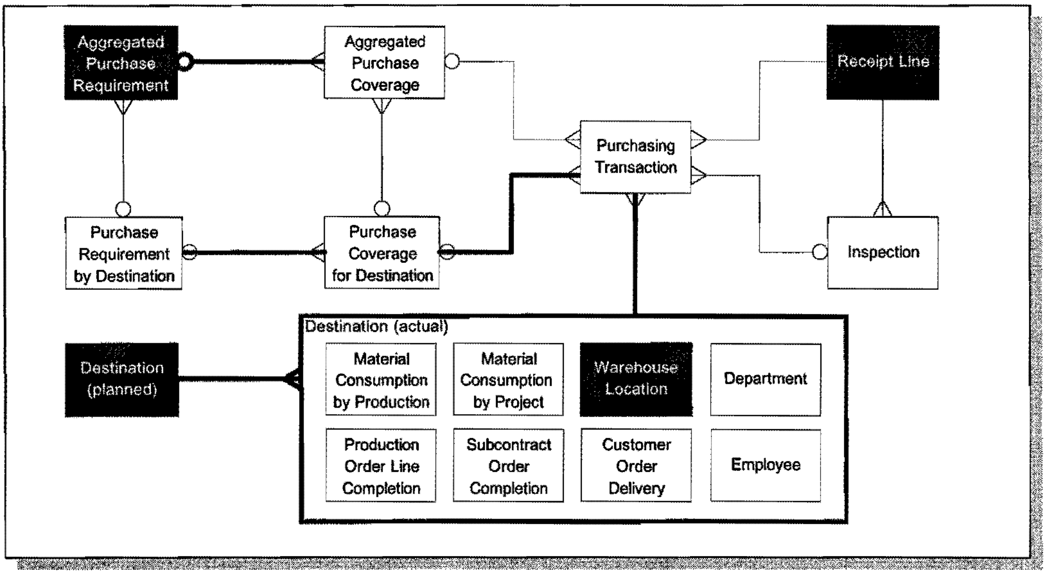


Figure 5.20 - Purchasing transactions

5290 The goods always have a destination, this can be a warehouse, an inspection location, a department, or an employee. With direct deliveries to customers (or other external destinations) the destination of this transaction will be a customer delivery. Of course this delivery involves the original sales or transportation order from which the purchasing requirement arose.

5291 The relationship between the *'Purchasing Transaction'* and the *'Purchase Coverage'* (and therefore the *'Purchase Requirement'*) indicates which part has already been realised or covered. In the *'Purchase Coverage for Destination'* the destination of the goods is predetermined, while in an *'Aggregated Purchase Coverage'* a explicit decision is required when the goods are received.

Invoicing

5292 The purchase invoice check is the financial completion of the purchasing process. Incoming invoices are primarily recorded in the accounts payable administration, after which they are available for checking by the purchasing function. After checking they are indicated as 'payable' within the account payable administration and payments can then be made. With the exception of the special situation that is described in paragraph 5276, i.e. 'pay on production' and variations on this theme (these invoicing methods are characterised by the lack of administrative activities, such as goods receipt and invoice checking).

5293 The left-hand side of figure 5.21 demonstrates that the normalisation of the n:m relationship types with the *'Purchase Invoice Line'* lead to a new *'Purchase Invoice Line'* on the right-hand side of the same figure. The latter can be found in most BIS. In practice this means that the *'Purchase Invoice Lines'* are usually created by the purchase invoice checking process.

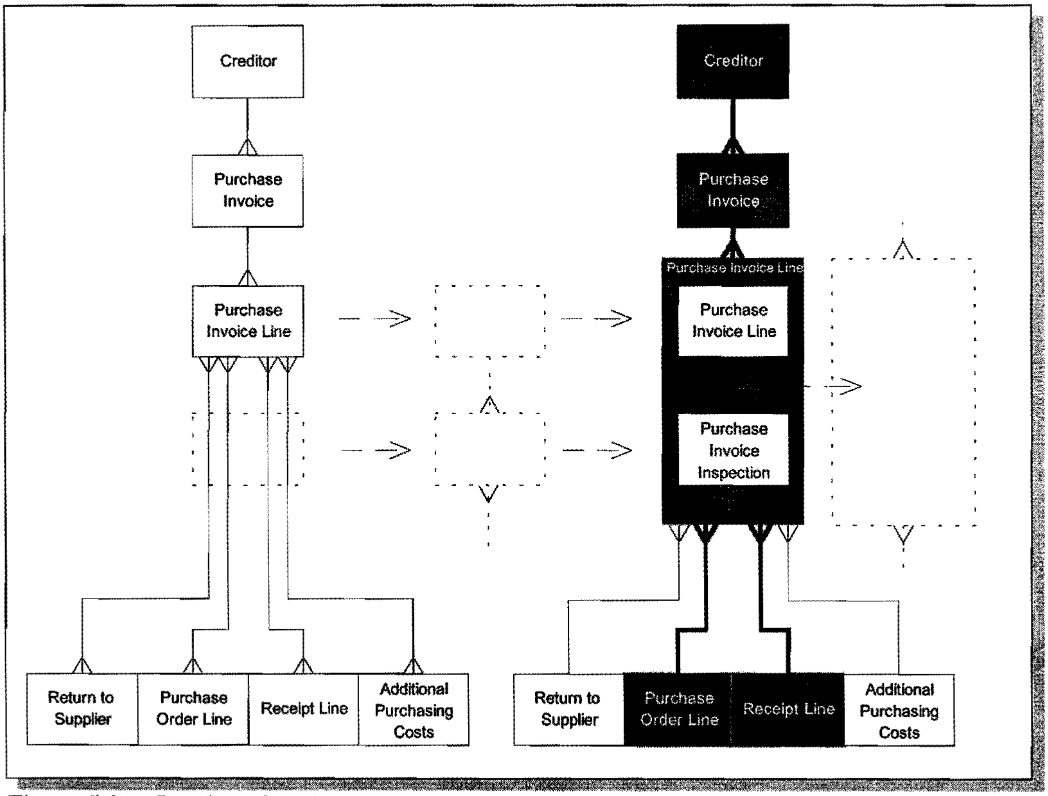


Figure 5.21 - Purchase invoice

5294 For invoice checking, the invoice is matched against the orders, the receipts, and the expected additional purchasing costs. Depending on the quality of the recorded data, systems with automatic matching are an option. A distinction is made between two types of invoicing:-

(a) post-invoicing based on the actual amounts received and the agreed purchasing conditions valid for the order;
(b) pre-invoicing based on the agreed conditions and quantity in the purchase order and not on the actual amounts and quality received.

Traditionally, this choice is often determined by the party (supplier or customer) who has the most power or who runs the most risks.

5295 For services there is usually no equivalent to the receipt. In most cases the receipt of the invoice is the trigger to assess whether the service was carried out in accordance with the agreement (the purchase order). Due to the lack of a receipt function, in the post-invoicing of services there only is a relationship between the purchase order and the invoice line, just as with pre-invoicing. This means there is no visible distinction in the data structure between pre- or post-invoicing for services.

5296 One invoice line may be related to several *'Purchase Order Lines'* or *'Receipt Lines'*. On the other hand, it is also possible that several (partial) invoices are received for one registered receipt line or purchase order line. The latter is normally undesirable and at most is useful for corrections on a previous invoice. The post- and pre-invoicing lead to the relationship types with the *'Receipt Line'* and *'Purchase Order Line'* respectively.

5297 Important considerations for invoice checking with post-invoicing are:-
(a) *the quality recorded.* If the received goods are still under inspection then the approval of the invoice has to be postponed;
(b) *goods return.* If returns are registered as negative receipt lines, there is a danger that the invoice is agreed; in fact the goods were received and because of the separate (negative) receipt line there is no relationship between the receipt and this return shipment;
(c) *partial deliveries.* Mostly invoice approval will take place once all the goods have been delivered;
(d) *changes.* For example, if the supplier sends slightly different items because the assortment has changed.

History and statistics

5298 A good purchasing history has to recognise two dimensions for each purchasing transaction (receipt) dealt with. First, there is a distinction in purchase data between the required, agreed, realised, and invoiced. Second these can be split up into the different purchasing conditions, such as the price, quantity, quality, and delivery time.

5299 Many systems define one or more aggregations next to the detailed history. These statistics are often company specific and mainly serve to provide reports quickly without having to examine all the historical details one by one (with the influence on the general response times). Basically, the aggregation is usually on supplier, purchase item, and period (year/month).

5300 The supplier's performance therefore is little more than a certain type of report from the purchasing history. To avoid having to calculate this every time, some standard aggregations (statistics) are usually kept for this purpose. Consequently the risk is that they are not satisfactory. It is essential to establish the relevant data in the purchasing history, as given in figure 5.22.

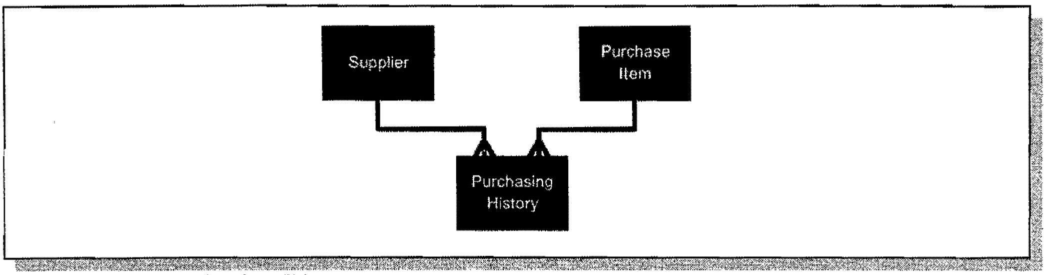


Figure 5.22 - Purchasing History

5301 It is useful to define which formula (or algorithm) is to be used to calculate supplier performance. This can be applied to the purchasing history.

Alternative items and suppliers

5302 If a required '*Purchase Item*' cannot be supplied by the preferred supplier, some alternative is required. Three alternatives can be distinguished:-

(a) an alternative supplier for the same purchase item;
(b) an alternative purchase item from the same supplier;
(c) an alternative purchase item from a different supplier.

5303 Though it is possible to consider these three kinds as separate data structures, the fact that they are interchangeable is reason enough to put them in one general structure. The foundation for this is the entity type '*Purchasing Assortment*', in which all relevant '*Supplier*' and '*Purchase Item*' combinations are listed. For such a combination several '*Purchasing Alternatives*' are possible, which in turn have to be part of the '*Purchasing Assortment*'.

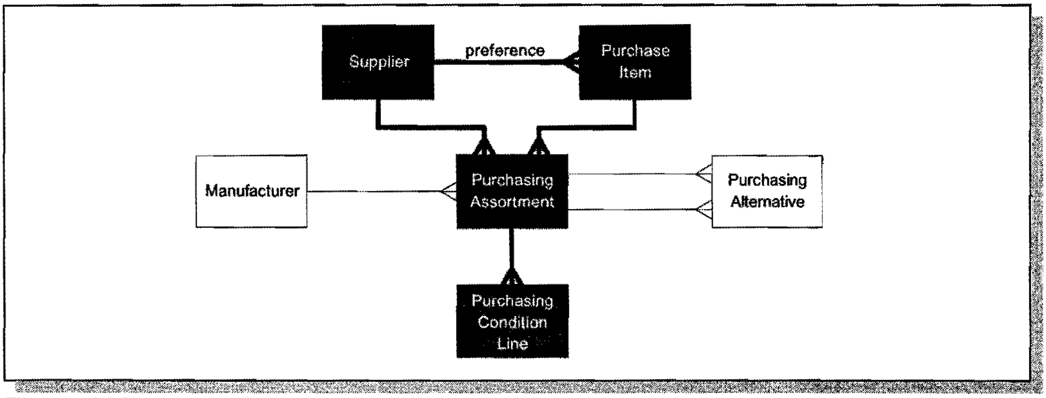


Figure 5.23 - Alternatives

5304 Potentially important attribute types with '*Purchasing Alternatives*' are the '*degree of preference*' for the supplier, and '*requirement percentages*', to maintain the proportion of mutual procurement between suppliers constant.

5305 Particularly for standard components that are purchased, it can also be useful to record the manufacturer of an item. If it is decided to purchase the items from a different supplier (trader), one can further refer to the original manufacturer. For example electronic components such as microprocessors.

Delivery addresses

5306 It would seem obvious that purchased goods are delivered to the purchasing organisation's receiving warehouse. However, it can also be desirable to deliver the goods directly to the customer (of the purchasing organisation). These kinds of deliveries are also called 'direct deliveries'. Another form is the delivery of goods to another supplier, for example when production activities are subcontracted. These kinds of deliveries are also called 'drop shipments'.

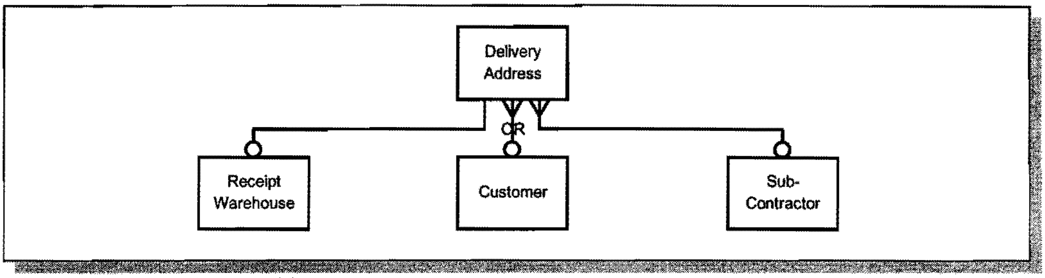


Figure 5.24 - Delivery addresses

5307 For 'direct deliveries', the '*Delivery Address*' referred to in the '*Purchase Order*' is the same as the '*Delivery Address*' of the customer, while for 'drop shipments' the '*Delivery Address*' is the subcontractor (another supplier).

Multi-site structures

5308 The way in which multiple sites (for example branches or plants) need to be supported by the information system depends greatly on the degree of decentralisation of responsibilities, and particularly the purchasing function.

5309 An advanced decentralised situation consists of fully independent sites, each with its own responsibilities and purchasing system. This situation can be easily implemented by means of separate implementations of the same system.

5310 A maximally centralised organisation is characterised by a central purchasing function in which all purchase agreements are made, recorded, and managed for the entire organisation. This is also relatively simple to implement using one system, where only several warehouses (and their delivery addresses) are distinguished. Here, non-warehouse specific purchase data is used.

5311 In most of the more complex organisation structures there will be some central data that applies to the entire organisation, and some decentralised data that applies to an individual site, including some decentralised management. Per entity type, independent of a specific company, it is not certain whether it should be maintained centrally or per site (or even a combination of the two, for example both central and decentralised purchasing conditions). This determines whether the attribute type '*site*' should occur in each entity type in question or not. If so, this attribute type is also part of the key attribute types.

5312 Theoretically, each attribute type can be site specific. This can result in the split of an entity type into two, i.e. the data (attribute types) applying to the entire organisation as one entity type and the site-specific data (attribute types) as a second entity type. This multi-site aspect is an example of a problem area that is still in development.

5313 In practice there are some preferred organisation types that can be recognised. In a more *decentralised purchasing function*, *site-independency* is useful for the following entity types:-
(a) *'Creditor'* and *'Supplier';*
(b) *'Standard Purchasing Conditions'*, which of course are already fixed per supplier;
(c) *'Blanket Order'*, the call-offs usually take place per site;
(d) *'Item'*, the extent to which an item is purchased or delivered to different warehouses will be determined per site.

5314 In the *centralised purchasing function* no more than the following entity types are expected to be *site-dependent*:-
(a) *'Ship From Address'*, as different sites have to be delivered from the closest supplier warehouse if possible;
(b) *'Purchasing Condition'*, when the transportation costs to the various sites differ substantially;
(c) *'Purchase Order Line'*, depending on the organisation per site, the ordering will often still be done locally (especially call-off orders);
(d) *'Aggregated Site Requirement'*, by definition the destination of the requirement is not fixed, of course the requesting site should be clear.

5315 Another kind of implementation than one complex data structure is also possible. Per site an independent system and data structure is implemented. The exchange and consistency of all data that apply to several sites, is maintained through automated message transfers, see also figure 6.28 and the explanation.

Controlling data

5316 In the previous step by step treatment, some aspects within the purchasing systems have not yet been dealt with. This mainly concerns matters regarding monitoring and management. These matters are described more generically below, i.e.:-
(a) expediting;
(b) texts;
(c) purchasing employees;
(d) change management;
(e) foreign currencies.

Expediting

5317 One important purchasing activity is monitoring the progress of expected deliveries, both regarding timing and quality. Depending on the risks and the complexity of the product, this activity varies from a simple registration of disparities afterwards, to the intensive tracking of supplier (production) activities. The latter, intensive form of monitoring progress is called expediting.

5318 In purchasing systems, and in data structures in particular, this activity is only minimally represented. The functionality mainly concerns:-
(a) monitoring progress per purchase order, for example on so-called expediting sheets. The data monitored here is rather company specific, implying the need for (extremely) extensive textual options at the level of the *'Purchase Order Line';*
(b) reporting expected delivery dates per supplier based on the last delivery date agreed upon. This information can be directly passed on to the supplier as a reminder. Additionally, it can also be considered as the potential work for the carrier in an active approach of the suppliers;

(c) the comparison of the cumulative delivery obligation with the cumulative production of the supplier in a 'just in time' delivery situation.

5319 However, it is important to continue to distinguish (in attribute types) between required, agreed upon, last known, and actual delivery date.

Texts

5320 For each entity type it can be useful to add extra text. This not only applies to purchasing. Thus, texts can add information about transactions, facts, and objects. A first distinction can be made between texts for internal and external use.

5321 The internal texts are intended for all direct users of the information system. In principle these can be found within the organisation. The texts are mainly informal. User friendliness for retrieval and definition are essential.

5322 External texts are printed on documents that leave the company, for example freight notes, order forms, or customs documents. These documents have a more or less fixed lay-out and the texts have a certain formal status. The following matters are important:-
(a) defining standard texts with possible options for alternations in a dynamic entity type like an order;
(b) accurate influence on the place and type of printing;
(c) identification of different texts used in the same entity type, for example specific texts for transport documents or a supplier item description;
(d) language dependency, depending on the destination of the document (supplier, transporter, or intermediary).

Purchasing employees

5323 For each function, and certainly for the purchasing function, it is important to establish the responsibilities of the employees involved. For example, which buyer is responsible for a certain supplier, purchasing group, or a purchase order that has been placed. In principle almost any entity type can have a relationship with the entity type '*Purchasing Employee*'. The most important ones are listed here.

5324 The static entity types that buyers are responsible for are:-
(a) '*Suppliers*';
(b) '*Purchase Item Group*' and '*Purchase Item*'.

5325 The purchasing activities can be found in the more dynamic part of the RDM. This means there are relationships with the entity types:-
(a) '*Purchasing Condition*';
(b) '*Purchase Quotation*';
(c) '*Blanket Order*' and '*Purchase Order*';
(d) '*Aggregated Purchase Requirement*' and '*Purchase Requirement by Destination*';
(e) '*Purchase Invoice Line*';
(f) '*Purchasing History*'.

Change management

5326 Depending on the status of the information, for example, if it has already been supplied to third parties, changes can be quite interfering. Naturally, the situation before the change has to be stored whatever the status. In principle this change management applies to each entity type.

5327 There are three options for implementation:-

(a) extending each entity type with a time and date stamp as a key attribute type. In this way a history can be built up within the entity type. The disadvantage of this solution is the extra load that every program function carries. *'Purchasing Conditions'* are typical entity types where this implementation is quite common. This is because the valid dates are known in advance and also because this data has to be available in the system well before the start;

(b) a shadow structure that is identical to the normal data structure, but with an extension for the time and date attribute types (see the previous implementation). The advantage is that the extra load no longer is present for most program functions, while nevertheless all changes can be stored. However, in this implementation there is a relationship between the original entity types and the shadow entity types;

(c) transaction logging in which each addition, alteration, or deletion is stored in separate entity types. The essential difference with the previous solution is the lack of a relationship between the original entity type and the entity type in which logging takes place. Such logging types are therefore often sequential.

5328 In each of the implementations described, attribute types can be added that tell us which user of the information system made the changes.

Foreign currencies

5329 In all places where financial amounts are recorded, it is necessary to indicate the currency. If this is not the case some standard currency will be used that applies to the entire legal entity or the whole organisation. For purchasing this particularly applies to the *'Purchasing Conditions'*, *'Additional Purchasing Costs'*, and *'Purchase Invoices'*.

5330 A conversion table (foreign exchange) is required to make the conversions. This conversion table is identified by the foreign currencies and a starting (validity) date. Here, a distinction can be made between the buying rate, selling rate, and middle rate. The rest of the data structure is trivial. The complexity of using foreign currencies lies in the conversion date and the type of rate being used.

5331 As far as the conversion date is concerned, there is a multitude of alternatives. Think of some typical moments in the purchasing process for example:-

(a) the moment at which the contract is signed;

(b) the moment at which the physical transaction takes place, so that the goods or services are delivered (not to mention the partial deliveries);

(c) the moment at which the financial transaction takes place, i.e. when the invoice is sent (or even received or paid).

5332 The data structure only shows whether foreign currencies are supported in the places mentioned, but not to which degree.

Specialised purchasing situations within the RDM

5400 The purchasing RDM has been generically described. This model therefore applies to various purchasing situations in different company situations. This means generic entity types are used, such as *'Purchase Requirement by Destination'*.

5401 However, the relationship with other systems and functions is essential in the assessment of purchasing systems. This is where purchasing systems differ from each other. Here, some specific purchasing situations are made explicit, which are hidden in the generic part of the RDM.

5402 The following situations will be discussed consecutively:-

(a) the co-operation between purchasing and production, the situations where goods are purchased specifically for production and subcontracting;
(b) the co-operation between purchasing and sales, aimed at customer-specific purchasing or subcontracting;
(c) the purchasing of transportation, a special form with a direct link to the purchase, sale, and the internal transport of goods between sites.

Co-operation between purchasing and production

5403 In the relationship between purchasing and production, three situations can be distinguished:-

(a) specific purchases for the benefit of production orders;
(b) subcontracting of production operations;
(c) subcontracting of complete production orders.

5404 Each of these purchasing situations is represented by a separate data structure. Note that these structures are no different than those described in the general RDM, and are only specialisations.

Direct purchasing for production orders

5405 In the production process use is made of raw materials, additives, and components that normally are held in stock. However, there may be reasons not to have these materials in stock well in advance because:-

(a) the increasing pressure to minimising stocks;
(b) the specialisation of the required materials making it impossible to know which exact specifications and requirements will have to meet.

5406 The procurement by the purchasing function will only commence once enough insight has been gained into the planned production. This still means purchasing may take place through general stock, i.e. without an explicit relationship between the purchase order and the production order. However, problems may arise if plans change and the number of orders is so large that it becomes hard to control them manually, because of:-

(a) changes in the production orders that have to be forwarded to purchasing all the time;
(b) alterations by the supplier that have an influence on the production planning process;
(c) the exact delivery of the proper materials for the right production orders, as soon as these have been received from the supplier.

5407 Maintaining the relationship between a requirement (being production in this example) and its coverage (by placing a purchase order) is also referred to as pegging. In full pegging the relationship is maintained on both sides, from the requirement to the purchase order that covers this requirement as well as from the purchase order to the requirements that are to be satisfied in this way.

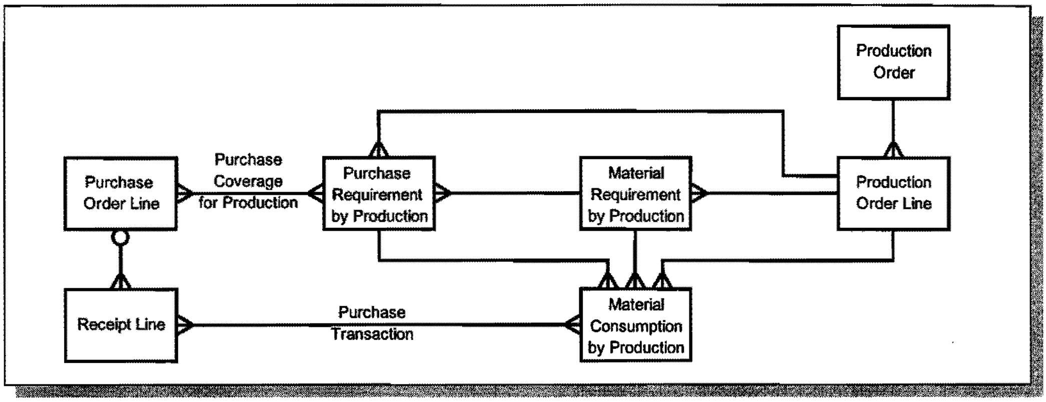


Figure 5.25 - Purchasing for production

5408 A brief explanation of the RDM specialisation shown in figure 5.25: a production order is built up of a number of operations (*'Production Order Lines'*), that produce the desired product. Per (planned) *'Production Order Line'* there is a material requirement for the different types of raw material, components and/or additives. The next step may be the decision to cover one of these requirements by purchasing the material, leading to a *'Purchase Requirement by Production'*. Instead of purchasing, a choice could have been made to fulfil this material requirement differently: by getting the materials from a different warehouse/site or producing it oneself. In principle one *'Material Requirement'* can even lead to several *'Purchase Requirements by Production'* due to a possible spread in time of this requirement.

5409 On the basis of *'Purchase Requirement by Production'*, purchase orders (or call-off orders) can be made. The receipt and use of these purchased materials then lead to the *'Material Consumption by Production'*. Alterations in the realisation (more/less received or used) leads to the complex (n:m) relationship type between the *'Receipt Line'* and the *'Material Consumption by Production'*.

Subcontracting production operations

5410 The difference between subcontracting and normal purchasing is the supply of the materials and production details by the purchasing organisation to the subcontractor. Without these activities, this type of purchasing could be treated in the same way as normal purchasing.

5411 Here, subcontracting from the shop floor will be discussed, where one of the operations (*'Production Order Lines'*) is carried out externally. Normally, previous and/or following operations are carried out internally. It is obvious that the subcontracted operations are relatively simple and standardised, for instance galvanisation.

5412 Subcontracting is a form of purchasing on the one hand with functions like supplier negotiation, agreements, receipt of goods, and purchase invoice checking. On the other hand, it can also be viewed at as a special form of production with operations that are carried out externally rather than internally. If subcontracting is only of secondary importance in a company, both implementation alternatives can be considered.

5413 Often, however, subcontracting requires more attention to details than ordinary purchasing or normal production activities. Material has to be supplied from previous operations or from stock to the subcontractor. Basically the required materials are defined in a separate level in the bill of material with a link to the routing where necessary, comparable to the normal internal operations. After receipt of the externally produced result, the production process is continued internally.

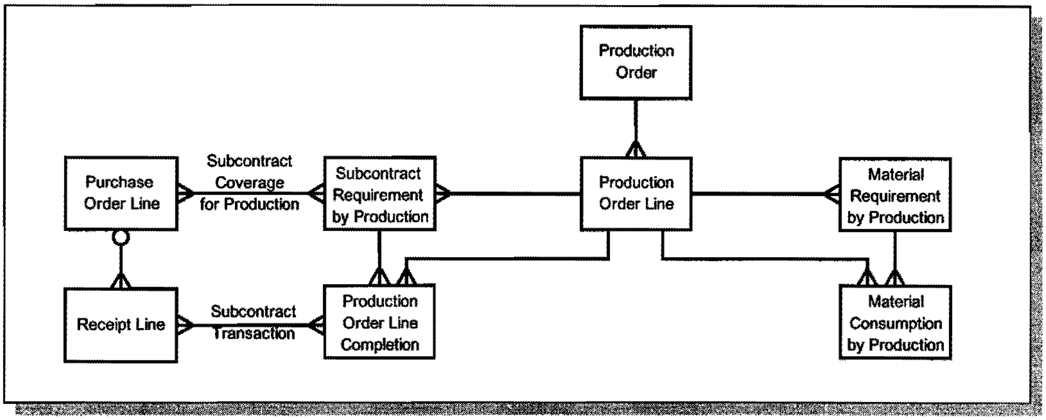


Figure 5.26 - Subcontracting production operations

5414 In figure 5.26 the RDM specialisation is shown. Comparing this to the structure of figure 5.25: the purchase requirement is now no longer a material requirement, but the direct requirement of a (planned) '*Production Order Line*', since the operations are being subcontracted. This purchase requirement leads to a purchase order (or call-off order) for the external execution of the operation. The required material should then be sent to the subcontractor. This material can be supplied from stock, supplied directly from the production line or be specially purchased. These relationships are found elsewhere in the RDM. The externally required materials are on the right-hand side of figure 5.26 ('*Material Consumption by Production*'). After receipt the processed or produced goods will continue in the internal production process. After a possible receipt inspection, this implies that the receipt is basically equivalent to the actual completion report of the '*Production Order Line*'.

Subcontracting of complete production orders

5415 A more complex type of subcontracting than that of individual operations concerns the subcontracting of complete production orders. No internal operations immediately precede or follow the subcontracted operations. This type is also known as subcontracting from the warehouse, in contrast to the previous type, which is also known as subcontracting from the shop floor. Still characteristic is that material is supplied to the subcontractor by the purchasing company, which makes it fundamentally different from normal purchasing.

5416 In figure 5.27 the subcontracting order can be regarded as a '*Production Order*' of a specific type. For clarity this type is specialised in the '*Subcontracting Order*'. The entire order is subcontracted and leads to a purchase requirement. Recording the material requirement per '*Subcontracting Order Line*' allows us to be more accurate with the delivery of material during the external production activities. This also requires feed back concerning changes in subcontractor plans.

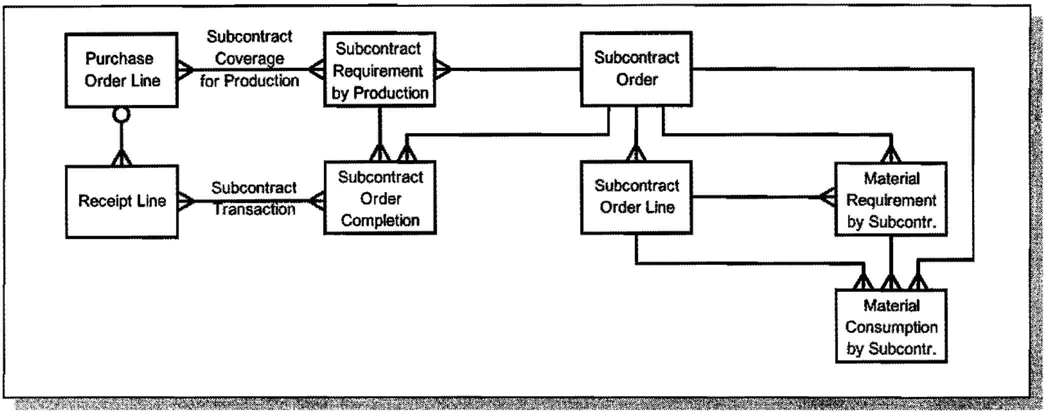


Figure 5.27 - Subcontracting complete production orders

5417 In principle this type of subcontracting can be supported with the previous data structure shown in figure 5.26. This can be done by defining a production order as a subcontracting order with only one production order line that is subcontracted. The only difficult part to implement is the delivery timing of the goods. Nevertheless in (production) practice there is a clear difference between both types of subcontracting, mainly because of the complexity.

Co-operation between purchasing and sales

5418 There are a number of situations where a direct relationship is required between the sales and purchasing function. Examples can be found in trading companies where items ordered by customers are directly purchased, often because they are so customer specific or for other reasons no stock can be kept. Valid reasons for not keeping any stock or only limited stock can be caused by the available storage capacity, the risk items becoming obsolete, the capital investment or simply a lack of stock giving rise to a back order.

5419 In these situations there is often no reason to receive goods into the company warehouse and then dispatch them to the customer. A 'direct delivery' in which the delivery address of the purchased goods is the customer delivery address is an obvious requirement. Here, two situations can be distinguished:-

(a) specific purchasing for a customer order;

(b) subcontracting customer orders, comparable to the previous discussion of subcontracting of production orders.

Purchasing for customer orders

5420 In specific purchasing for customer orders, the requirement is defined in the customer order (line) that is to be covered by purchasing. For this purpose several customer order lines can be combined into one purchase order line. Alternatively, one customer order line can be split into several purchase order lines, for example if they are spread across several suppliers to reduce risks.

5421 The receipt can lead directly to a (partial) customer delivery, possibly combined with other ordered goods to form one total shipment. A *'Receipt Line'* can be intended for several destinations, so repackaging might have to take place. Furthermore, goods can be delivered directly without the trader having seen them, in which case the 'Receipt Line' is identical to the *'Customer Delivery Line'* (1:1 relationship type).

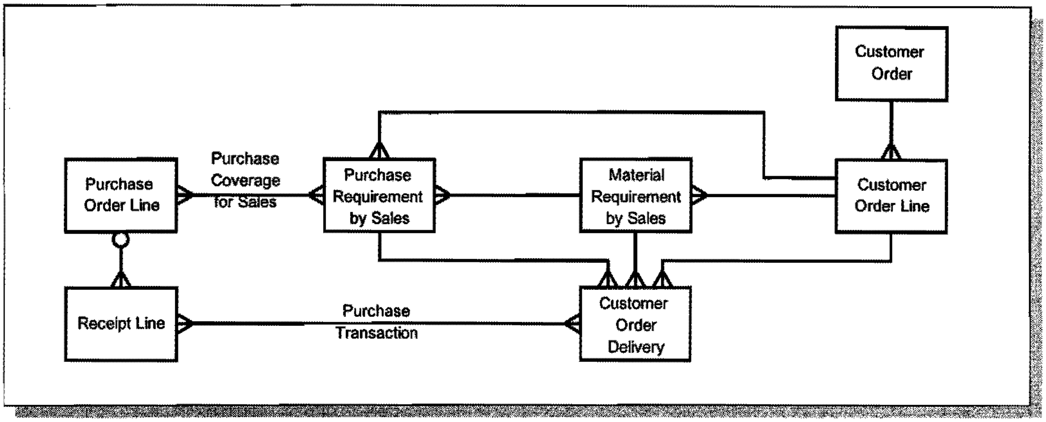


Figure 5.28 - Purchasing for customer orders

5422 Specific situations usually are simpler: for example a *'Customer Order Line'* may lead to just one *'Purchase Order Line'* or it can be fulfilled from stocks. The *'Receipt Line'* then also is the delivery to the customer. This should be taken into account when looking for a suitable BIS, i.e. choose a system that supports what is necessary, not the most complex situation.

Subcontracting customer order

5423 Comparable to the subcontracting of production activities as described earlier, direct subcontracting of a customer order can also take place. This means that no production activities are carried out under one's own management, but that material is supplied to a third party, who delivers the end product. On the basis of the customer order (line) recorded, subcontracting is organised as a purchasing activity. The materials to be delivered are prepared and dispatched. After external processing (for example assembling) the final products are received and sent to the customer. If necessary direct shipment to the customer can take place from the subcontractor ('direct delivery').

5424 This type of subcontracting can also be realised by making a production order first and then organising the subcontracting, see the paragraph 5415 on subcontracting of complex production orders. The direct method described here is not only shorter, but requires fewer activities and does not use any unnecessary production functionality. A common application is in trading companies that do not have any production facilities of their own.

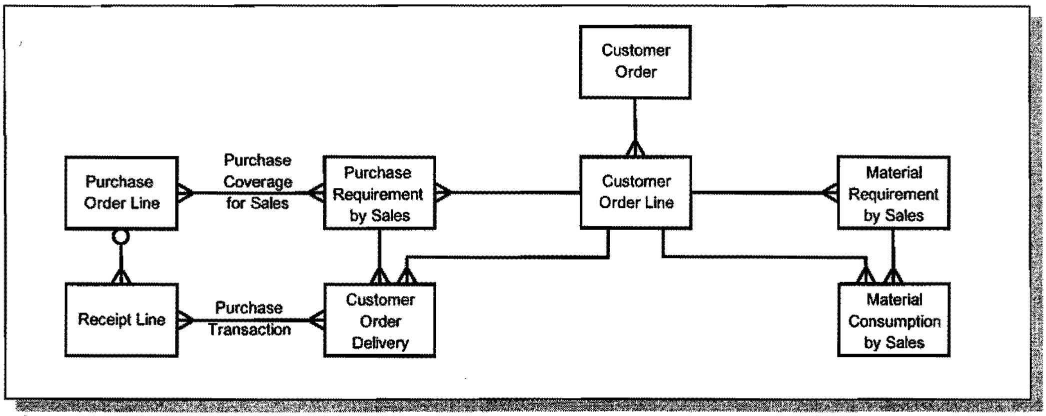


Figure 5.29 - Subcontracting customer orders

5425 In figure 5.29 the '*Material Requirement by Sales*' is shown on the right side, which describes the materials to be delivered. These requirements originate from the bill of material that is linked to the sales item. The '*Material Consumption by Sales*' describes the eventual actual use of these materials.

Purchasing of transportation

5426 The subcontracting of transportation is a special kind of purchasing. This service can be related to transportation for:-
(a) the purchasing of goods, i.e. deliveries 'ex-works';
(b) the sale of goods, i.e. carriage paid deliveries;
(c) the direct delivery of goods as described before, in which the goods are transported directly from the supplier to the customer;
(d) the internal transportation of goods, i.e. from one storage location to another.

5427 The following specific data is relevant for subcontracting transportation with respect to other types of purchase orders:-
(a) the carrier (i.e. the supplier of this service);
(b) the route (i.e. the purchase item that is being purchased);
(c) the goods, the quantity, and the kind of packaging (type of container, full truck load, tonnage, etc.);
(d) the ship from and delivery addresses;
(e) the departure and arrival times (including times at which loading/unloading is/is not possible);
(f) the costs (and other purchasing conditions);
(g) the required documentation;
(h) the actual times / costs / accounting of the transportation that took place.

5428 A large amount of this data is very specific for the kind of transportation used (by air, by land, or by sea), the kinds of goods (bulk, general cargo), the scope (international, national), and combined transports. Much data (such as the goods to be delivered, the addresses, and the times) follow from the relationship with the purchase order, customer order, and the internal transfer order for which the transportation is to take place. As a result, they are part of the specification of the transportation to take place.

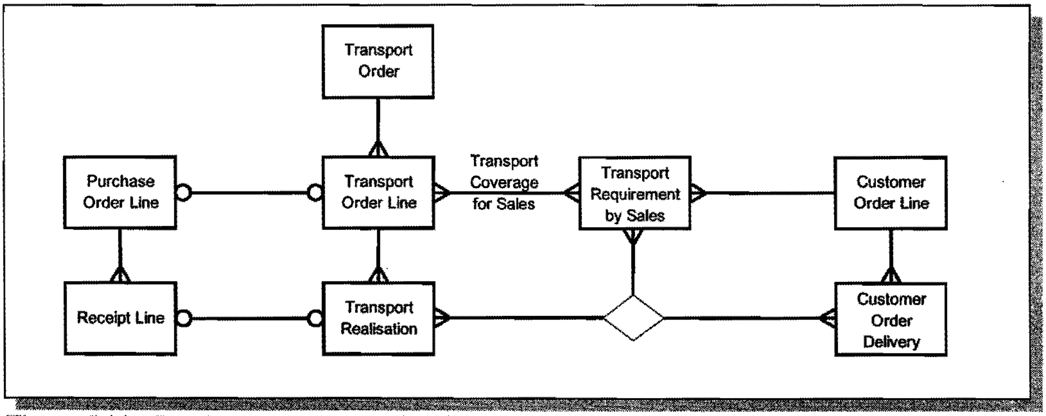


Figure 5.30 - Purchasing transportation for normal customer orders

5429 In the implementation the transportation order can be viewed as a special type of purchase order, comparable to a purchase order for '*Assortment Items*' or for '*General Services*'. In particular, the relationship with another purchase order, customer order, or other internal transfer order is of great importance. This type is implicitly modelled in the RDM that was described in the previous chapter. The specialisation is hidden in the purchase item, the requirements and realisation per destination.

5430 In figure 5.30 this transport order is depicted as a specialisation of a purchase order. The transport requirement arises from the customer order. In the next figure 5.31, a similar description is given of situations in which the transport requirement is the result of a purchase order, a 'direct delivery', or an internal transfer of goods. As a result of these '*Transport Requirements*' one or more '*Transport Orders*' are made.

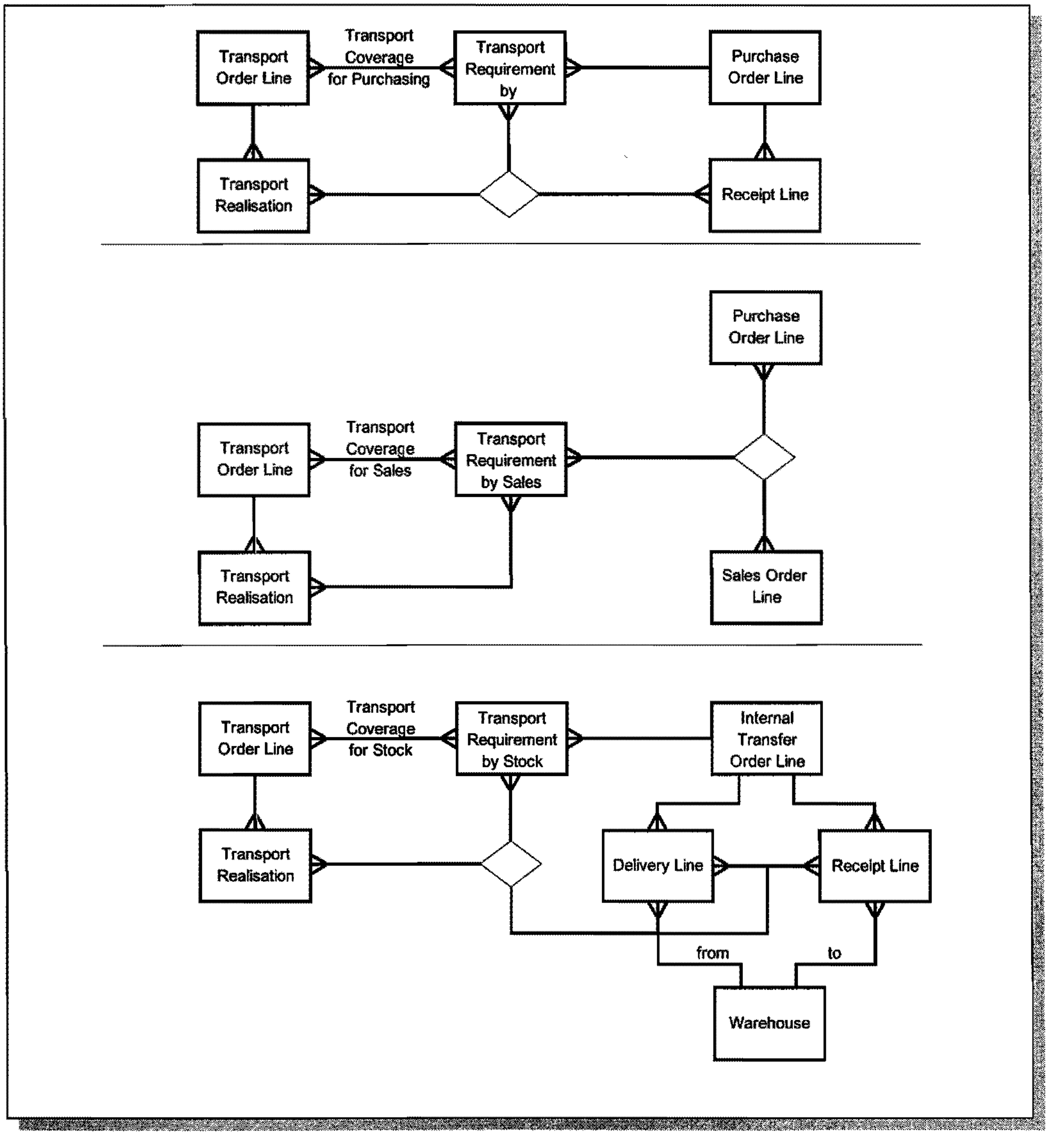


Figure 5.31 - Transportation for goods 'ex-works', 'direct deliveries' and internal transfers

5431 A similar situation arises if the *'Transport Order'* is carried out using one's own means. The typical purchasing aspects such as the financial administration and the purchase invoices checking, are taken care of via the related purchase order if transportation is subcontracted.

5432 In systems not specifically aimed at transportation, such as production control software, this model is given very limited support. Usually, not more than the type of costs for transport can be recorded, for example as part of the sales costs in a customer order (with for that matter a very limited tariff structure) and some extra data can be included in a customer order concerning the carrier and a route code. Usually the actual integration with the transport purchasing function is lacking. This implies that the explicit issue of transport commissions, the confirmation of the transportation, the financial administration, and the transport invoice check are also lacking. If these kinds of integrated systems are to be interfaced with specific transportation software, alterations in the transportation requirements (customer orders and plans change), and feedback after the realisation of transportation need to be taken into account.

Summary of characteristics of purchasing software

5500 It is striking that the discriminating characteristics of standard software gradually shift as a result of the growing requirements of organisations. Whereas until recently the support of blanket orders and purchase invoice checking were viewed as important distinctive characteristics, we now see that nearly all BIS of a reasonable size provide this functionality.

5501 As a result of this data-oriented analysis, a number of characteristics that are used as selection criteria in practice have not become properly visible. A number of them were already discussed in the RDM. Some of the characteristics that are not visible or only poorly (depending on the choices made in the implementation of the functionality) are:-

(a) the distinction between delivery schedules and call-off orders. As was noted it is possible to use the same data structure for this. Yet most BIS have chosen for a separate implementation so that this difference often is shown in the data structure;

(b) the support of the rolling forecast by means of planned purchase orders or on the basis of delivery schedules. This is partly because a fully crystallised 'best practice' implementation for this functionality does not yet exist;

(c) special types of contracts, for example:-
    (i) standing order (regular deliveries until further notice);
    (ii) fixed frequency deliveries;
    (iii) fixed quantity deliveries;

(d) the distinction between standard and continuous purchasing conditions. Also because this distinction is not made in systems;

(e) the time components contributing to the purchase lead time (these are typical attribute types of the purchase item);

(f) more accurate conclusion about which purchase items (see figure 5.5) are and are not properly supported (systems implement nearly everything via an item master file with many hundreds of attribute types. However, this does not mean that everything is possible and to draw conclusions the attribute types have to studied in detail);

(g) concrete support for communication (and feedback!) with suppliers, with the possible support of EDI;

(h) some modern payment mechanisms, like 'pay on receipt' and 'pay on production'.

5502 Most discriminating characteristics of systems clearly stem from the RDM described. The most important ones are summarised below:-

(a) complex supplier's organisation with a distinction between:-
  (i) the party sending the invoice (creditor);
  (ii) the party with which price agreements have been made / the contract has been signed (contract address);
  (iii) the party where purchase orders or blanket order call-offs come from (ordering address);
  (iv) the location from which goods can be collected (ship from address);
  as well as their mutual relationships.

(b) which purchase items are explicitly supported (i.e. not by means of a general solution, like the purchasing of transportation by defining separate items with a price per route). Here, particular interested is paid to:-
  (i) special items without them having to be recorded in an item master file;
  (ii) variant items;
  (iii) transportation services;
  (iv) subcontracting operations and routings;
  (v) hiring of production capacity;
  (vi) services in general.

(c) distinguishing a higher level in purchase items, '*Purchase Item Groups*' on which conditions, prices or contracts can be agreed upon, for example a price for the item group 'paint'.

(d) the support of (reusable) packaging material / purchase bills of materials / combined delivery goods, whose functionality is partially overlapping.

(e) special purchasing conditions, such as:-
  (i) quality conditions;
  (ii) service conditions;
  (iii) ownership conditions;
  (iv) conditions for returns.

(f) quotations in which a special distinction is made between the request for a quotation sent to several suppliers and the reactions (quotations) from the individual suppliers.

(g) delivery schedules with a feedback mechanism for exceptional situations.

(h) returns to the supplier:-
  (i) related to a receipt;
  (ii) related to a purchase order;
  (iii) stand alone, because of surplus or obsolete stock.

(i) a batch or lot number is already known and has to stored when a purchase order is placed or a blanket order call-off is made.

(j) arranging and controlling transportation of deliveries 'ex-works'.

(k) direct purchasing (without decoupling via a stock point) for:-
  (i) production orders;
  (ii) customer orders / customer order call-offs and 'direct deliveries';
  (iii) projects;
  (iv) purchase orders (when dealing with subcontracting).

(l) pre-invoicing, where invoice checking (and payment) take place before the registration of the receipt.

(m) a purchasing history split up into required, agreed, realised, and calculated data elements. This data may include purchasing conditions, namely price, quantity, quality, and delivery time.

(n) alternatives:-
  (i) an alternative supplier for the same purchase item;
  (ii) an alternative purchase item from the same supplier;
  (iii) an alternative purchase item from a different supplier.

(o) multi-site, which data is or is not site-dependent.

(p) foreign currencies and especially which exchange rate and value date is used for conversion.

6 An effective Reference Data Model for State-independent Manufacturing Data

6001 The area of manufacturing is of a different nature from purchasing as described in the previous chapter. Whereas purchasing is (still) mainly aimed at transaction processing and administrative aspects, with manufacturing the product structure, production process, control, and management are the key issues. These structures specifically differentiate the large production organisations.

6002 Below from paragraph 6100 onwards a general outline will be given of 'State-independent Manufacturing Data'. Following the functional outline, choices concerning state-independent data will be discussed. In addition, an indication will be given of the production typologies and situations that are in principle included in the modelling and assessment.

6003 From paragraph 6200 on, the actual Reference Data Model (RDM) will be described in detail. Considering the complexity of the model, the explanation is built up in a large number of steps. The projections of the data structures of four BIS demonstrate on the one hand that projection is possible, and on the other that the result clearly has a discriminating value for the various BIS.

Scope of 'State-independent Manufacturing Data'

6100 The area of manufacturing contains an important part of the heart of a production organisation with all its inherent complexity and diversity. Therefore, a complete description embracing all situations is impossible. The Reference Data Model (RDM) is largely based on present knowledge and the 'state of the art' in this area, which is stored in various production control and shop floor control systems. Besides the projected data structures given in paragraph 6200 and following, a great many other systems were studied in the development of the RDM. The challenge taken on in drawing up the RDM involved the structuring and combining of a multitude of philosophies and implemented solutions. Here, use was made of much practical experience to enable the right choice to be made between including details for importance and completeness. Besides the BIS on the market some important external sources were used:-
(a) research commissioned by ITC on Shop Floor Control systems [ITC95];
(b) the book 'MRP in Process' [VLM93];
(c) the book 'Computerisation in stock management and production control' [LEE92];
(d) the book 'Business process engineering: Reference models for industrial enterprises' [SCH94].

6101 The chosen scope of the area of 'State-independent Manufacturing Data' can be explained from three perspectives:-
(a) the distinction between state-independent versus state-dependent;
(b) the functional position;
(c) the typologies.

6102 While defining *state-independent data* use is made of the categories defined in figure 4.12. Basis on this categorisation, the RDM is restricted to the standing data and the associative data, see figure 6.1.

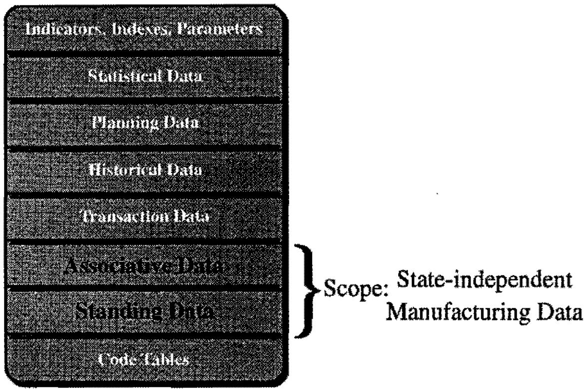


Figure 6.1 - Scope of state-independency

6103 The standing data is the basic entity types that often can be retraced physically and directly in the organisation. Typical examples within manufacturing are materials, machines, tools, customers, factories, departments, and personnel. The associative data are formed by the relationships between this standing data and between other associative entity types. Typical examples are bills of material, routings, alternative materials or operations, and customer-specific bills of material.

6104 The various kinds of orders, such as planned or issued production orders and purchase orders, fall *outside the scope*. This is also the case for the various (production) plans, such as the 'master production schedule' (MPS), and also for the customer-order-specific bills of material, and the allocation of material or other resources to sales and production orders.

6105 *Functionally*, a number of levels can be seen in production and control, see figure 6.2. Here, the scope is also shown from this perspective. As the scope is restricted to the state-independent data, there is a major overlap between the various control levels. Thus, bills of material and routings occur at all levels, be it with more or less detail

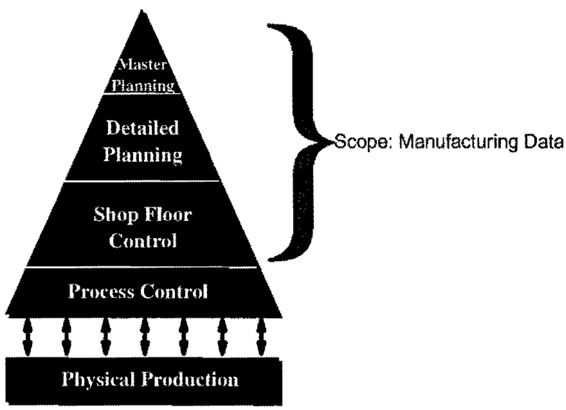


Figure 6.2 - Scope of manufacturing data

6106 There are a number of related areas overlapping the 'State-independent Manufacturing Data', but for the rest fall outside this scope. For the sake of clarity the related areas are:-

(a) *Product Data Management (PDM).* These systems are aimed at all data and functionality concerning product development, as well as the management of product documentation. The overlap which is part of the RDM includes the product characteristics relevant for production, such as the product structure, configuration, and classification of products;

(b) *Computer Aided Design (CAD).* These systems are focused on the design of products production processes. The overlap lies in the resulting specifications, the bills of material, routings and such. Furthermore, the engineering changes should be supported within the manufacturing system too;

(c) *Supervisory Control and Data Acquisition (SCADA).* These systems are aimed at the control of individual machines and production processes, and are part of the process control level in figure 6.2. The overlap lies to a great extent in the standing data required for control, like machine, process and product characteristics, which are essentially part of the RDM.

6107 The overlap with such systems leads to various links between BIS and stand alone packages for PDM, CAD, SCADA, and also, for example, finite capacity scheduling. These systems are increasing in functionality, and thus the overlap increases. Also the BIS are developing such that these links are becoming more intensive, leading to full integration.

6108 A third perspective is based on the *typology.* In the literature, a multitude of typologies is described with respect to product characteristics, production process, organisation and control. An effort is made to visualise all these characteristics in a (usually two-dimensional) matrix, which does not lead to actual understanding. Here, these different viewpoints are separated and explained one by one. Basically all concepts described within this typology fall within the scope of the RDM.

Product typology

6109 A simple division can be made using the *nature of a product*:-

(a) *discrete products,* that are measured in discrete units, like pieces;

(b) *continuous products,* that are measured in continuous units, like length or area (for example sheet material);

(c) *bulk products,* that are measured in volume units, such as litres or kilograms. Such bulk products must be stored in storage tanks and can not be 'dropped on the floor'.

6110 Naturally, the nature of the raw materials and semi-finished products may differ with respect to the finished products within one production process. Additives (materials) can also be categorised in this way.

6111 Another product division can be made via the *specialisation of the product*:-

(a) *standard products*, that are supplied to several customers or are used within production orders for several customers;

(b) *customer-specific products*, both customer and customer-order-specific finished products, semi-finished products, components and raw materials. Examples are variant items, configured items, and specials. These products are found in certain 'make to order' and most 'engineer to order' situations.

6112 For that matter, sometimes this division implicitly suggests parallels with concepts like 'make to stock' for standard products, and 'make to order' or 'engineer to order' for customer-specific products. However, this does not have to be the case, there are many production situations in which standard products are produced to customer order, and situations in which customer-specific products are stocked.

Process typology

6113 The production process can be categorised by the batch-size (see for example Botter [BOT80]):-

(a) *single items*, where (customer) specific products, expensive products or complex production processes are concerned;

(b) *small series*, where the emphasis lies in change-over times (and their reduction) mostly concerning customer-specific items, but also in typical JIT/Kanban situations;

(c) *large series*, where concepts such as rate-based manufacturing or repetitive manufacturing are concerned. Among other things, this typically comes down to production without production orders;

(d) *continuous production*, where the input and output-oriented process control (push/pull) and the accompanying maximisation of the output is concerned.

6114 Another production process categorisation can be made via the *physical organisation* of the production process:-

(a) *flexible manufacturing* (machine configurations or assembly locations), in which the production facilities are organised temporarily for one order or project;

(b) *job shop*, in which the production is physically grouped around similar production facilities and all products follow a more or less complex route through the factory;

(c) *flow shop*, in which the production is physically organised around the production process; the products follow a sequential production line, and only some production steps might be skipped;

(d) *process manufacturing*, in which production is similarly organised around the production process, but this organisation includes physical links (pipe lines and storage tanks) between the various process steps, usually due to the nature of the bulk product (for example fluids).

6115 Yet another categorisation of the production process can be made using the *structure of the production process*:-

(a) *sequential*, in which a product is processed step by step from a basic material to a finished product. A typical example can be found in the process industry and the (distillation) production of alcohol;

(b) *convergent*, in which raw materials and components are combined into semi-finished products that are in turn combined to make finished products. A typical example is the assembly industry;

(c) *divergent*, in which proper co- and by-products arise during the production process, to then be (re)processed. A typical example involves separation processes, such as selection on the basis of particle-size, used in the production of talc as a raw material for the paint industry;

(d) *network*, where complex, mostly project-oriented production processes are concerned and control is aimed at the individual production order. A typical example is true 'engineer to order' production, as can be found to an extreme degree in the space industry.

6116 This typology sometimes leads to classification problems, because it is not always possible to make a clear distinction. In many situations use will be made of additives (thus a sort of convergent structure) and waste products are also released (thus a sort of divergent structure). The combination of a convergent and divergent structure is of course a network structure that can be found in a number of semi-process industries. However, the differences become apparent immediately when the actual product and process structure is defined.

Controlling typology

6117 The categorisation of the controlling typologies is based on the *controlling concept* governing the production process:-

(a) *project-oriented*, in which an emphasis is placed on mutual balancing and communication within a complex structure of processes in a temporary configuration (the project);

(b) *cyclical*, in which setup, change-over and clean-up times are minimised by means of a fixed production sequence. The variation to be managed is in the lot-sizing of each individual production order. One typical example is the dairy industry;

(c) *MRP II*, in which the material and capacity requirement is exploded to individual orders via a hierarchical (product and process) structure;

(d) *JIT/Kanban*, where an exact requirement (in quantity and time) is produced with an optimal lot-size of one item (as a sort of supplementary order) without the daily support of a complex information system;

(e) *Optimised Production Technology (OPT) or Theory of Constraints (TOC)*, in which control is focused on the maximum use of bottle-neck machines or other constraints.

6118 This summery is by definition incomplete and only indicates the most important philosophies mentioned in the literature.

6119 Finally, another more frequently used categorisation is related to the control of the *customer order decoupling point*, the distinction between the stock-controlled and the customer order controlled part of the purchasing, production and distribution processes:-

(a) *make & ship-to-stock*, in which the goods are produced and distributed to decentralised storage and distribution locations without direct demand for customers (anonymously);

(b) *make-to-stock*, in which the goods are still produced anonymously, but distribution from this general stock is triggered by a customer order;

(c) *assemble-to-order*, in which the stock-controlled components are used to produce finished products on customer order;

(d) *make-to-order*, in which the complete production process is controlled by the customer order and use is made of stock-controlled raw materials;

(e) *purchase & make-to-order*, in which most raw materials are purchased on customer order, because of their speciality or due to the high costs involved.

6120 Though they are sometimes mentioned together, this last form is different from the 'engineer to order' situation, which says more about the customer (order) specification of the finished product than about the orientation of the customer order decoupling point. In this last classification stock-controlled components and/or raw materials may still be concerned. Even a combination with 'make to stock' is possible, if the engineered finished product is (regularly) delivered from stock after the customer-specific engineering process (which occurs once only). Engineer-to-order therefore is a combination of the typologies described earlier, in which the emphasis usually is placed on customer-specification, single item production (or small series), flexible manufacturing, network production process, project-oriented control and (purchase &) 'make to order' production.

6121 Some combinations of the above typologies are mutually exclusive and others do not occur in practice. However, a clear trend can be observed in the direction of combined production situations: hybrid production. Thus, many industries produce partially for stock and partially for customer order. Practice often is some production lines producing the voluminous (standardised) products based on repetitive manufacturing and the high variation customer-specific products are produced on a separate flexible production line.

Design of the RDM for 'State-independent Manufacturing Data'

6200 Here, the actual Reference Data Model (RDM) for 'State-independent Manufacturing Data' is worked out in detail. Due to the size and complexity it is built up in various steps with explanations, examples, and comments with regard to the implementation of BIS. This RDM is basically made up of five parts:-

(a) *material and capacity requirements* (from paragraph 6202). This is the heart of every manufacturing BIS and can be found under the names bill of material, recipe or production model;

(b) *usage and control* (from paragraph 6224). This contains subjects like a typology of items and bills, standardised (item-independent) requirements, alternatives, variants, classification, and multi-site;

(c) *relation to purchasing and sales* (from paragraph 6281). This covers the (state-independent) links with these areas, such as subcontracted bills of material and customer-specific bills of material;

(d) *typology of capacity resources* (from paragraph 6295). This concerns capacity resource groups related to individual resources, two-dimensional resources, skills and technologies, and machine configurations;

(e) *specialised / complex manufacturing* (from paragraph 6314). Here a number of subjects are discussed concerning the special characteristics of some production processes, such as divergency, multi-level operations and tasks, network structures of operations, sequence-dependent change-over times and qualities.

6201 In the development of this chapter use was made of Caubo's graduation work [CAU94], which was performed as part of this research. Where possible the terminology used conforms to that of the APICS dictionary [API92].

Material and capacity requirements

6202 In a number of steps the RDM that determines the state-independent material and capacity requirement of an *'Item'* will be constructed. The entity type 'Item' is defined as „Any unique manufactured or purchased part, material, intermediate, subassembly, or product" [API92], and is considered here synonymous with material, product and article. An extensive typology of this entity type is given from paragraph 6226 onwards.

6203 As a first step, figure 6.3 shows the traditional way of recording the material requirement: the multi-level *'Bill of Material'*. This is defined as „A listing of all the subassemblies, intermediates, parts, and raw materials that go into a parent assembly showing the quantity of each required to make an assembly" [API92]. An extensive typology is worked out in paragraph 6230. The synonyms *'Recipe'* and *'Formula'* sometimes used, have been given a broader definition in the RDM to clarify the intuitive difference.

6204 In the parent-component relationship all the required component items are determined, including the required quantity for the production of the parent item. As a component item can in itself be a parent item, we find the multi-level structure that can explode until the component items no longer are *'Manufacturing items'* but *'Purchase Items'*. There are other types of items, such as phantoms, by-products, product families, and generic items; this typology is worked out in more detail from paragraph 6226 onwards. Characteristic of the product structure shown in figure 6.3 is that it is convergent (hierarchical) and that the material requirement is determined by a so-called 'bill of material explosion'.

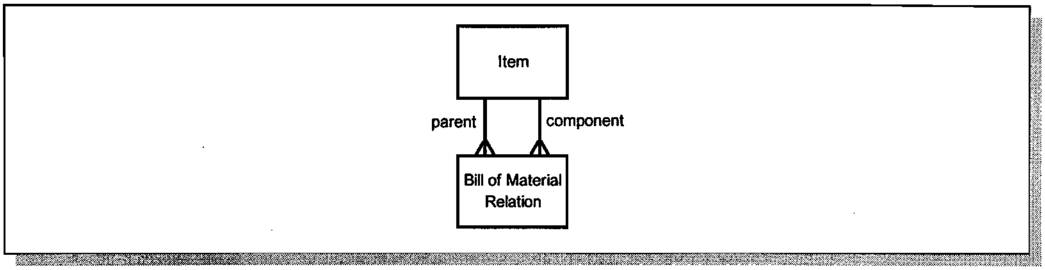


Figure 6.3 - Material requirement (multi-level bill of material)

6205 The process industry does not usually refer to a *'Bill of Material'*, but to recipes or formulas, but this hierarchical material requirement structure can be found in most systems. One of the few exceptions concerns systems based on implicit product structures defined in a 'production model' that are also divergent, for example, Prism and Renaissance. These are dealt with from paragraph 6316 onwards. In customer order controlled production it occurs regularly that components are not known beforehand. In the data structure this means that the component relationship type is optional (instead of an optional relationship type this is often implemented by means of a dummy item to which more data can be recorded such as the final date for specification,).

6206 Figure 6.4 shows the capacity requirement for the production of an *'Item'*. An extensive typology has also been defined for the different sorts of capacities, see paragraph 6295 and following. The systems used most frequently, however, go no further than this generic definition. Some systems (like Prism) have an even more abstract level, i.e. the *'Resource'* where non-durable and durable resources (*'Items'* and *'Capacity Units')* are indicated.

6207 A *'Routing'* is defined as „A set of information detailing the method of manufacture of a particular item, including the operations to be performed, their sequence, the various work centres to be involved, and the standards for setup and run times. It may also include information on tooling, operator skill levels, inspection operations, testing requirements" [API92].

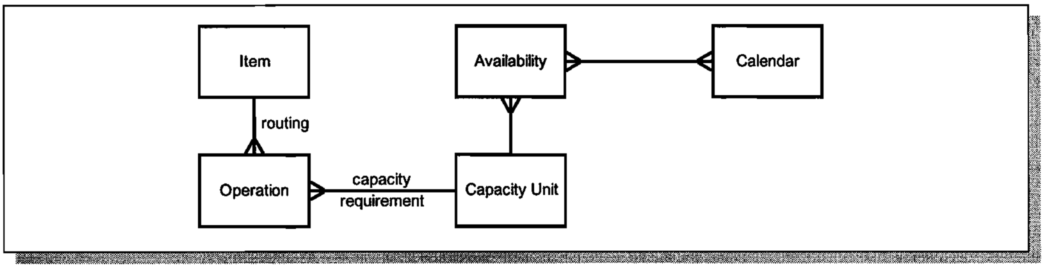


Figure 6.4 - Singular capacity requirement (routing or bill of operations)

6208 The entity type *'Operation'* is defined as „A description of an activity to be performed, normally contained in a routing document that could include setup instructions, operating instructions (feeds, speeds, heats, pressure, etc.), and required product specifications and/or tolerances" [API92]. Typical characteristics of *'Operations'* are the processing time and the setup time. The availability of the *'Capacity Unit'* is determined by means of a calendar. There are three variations for recording the availability:-

(a) the number of available hours per defined period (this division of time periods is usually fixed for the entire system). This variant (that predominantly occurs in older systems) is called 'bucketed' and has the advantage that calculation times are drastically reduced, in contrast to the other methods;

(b) the number of available hours per day, which, in fact, is still a bucketed calendar, only with the buckets of a day. Sometimes availability is also defined in a second dimension in the *'Capacity Unit'* in this variant, such as the number of units, for example: tools. From paragraph 6304 onwards further details on these two-dimensional resources is given;

(c) a time-phased availability, in which changes in the availability are established by means of starting and ending points in time. One example concerns tools that are purchased or produced and are available from that moment on. This variant is further elaborated from paragraph 6308 onwards.

6209 In figure 6.5 the previous forms are combined, which implies that actual differences can only be found in the attribute types of these entity types. Altogether this gives us a basic structure that can be found in the many traditional MRP-systems.

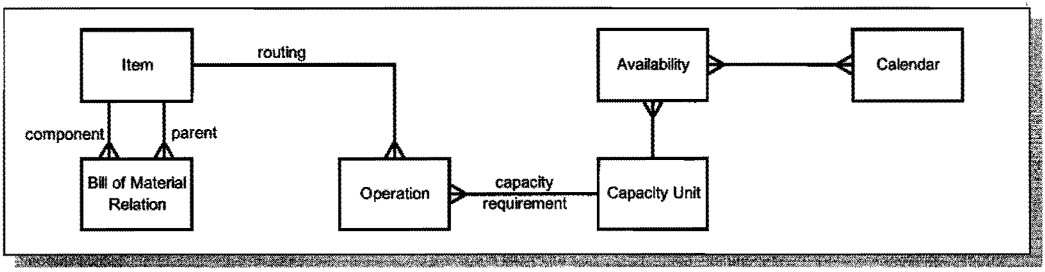
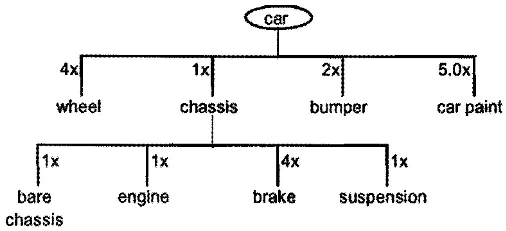


Figure 6.5 - Simple combined material and capacity requirement

6210 Figure 6.6 shows an example of this structure. The identified attribute types are underlined in the table headers. The example is naturally an extreme simplification of the requirement structure.

6211 In all these structures the material and capacity requirement was decoupled. The required materials can only be reserved at the beginning of the production process, the first *'Operation'*. For more lengthy processes this means that ordering or reservation of stock may occur too early and thereby unnecessary monopolisation of material and capital. In some systems this is solved with a so-called *'lead-time offset'* in the *'Bill of Material Relation'*. This is a fixed time after which the material reservation takes effect. However, this is not enough to deal with variable run times (for example, through interruptions, complex planning, or stochastic queue times). If there are long and variable running times, then it is better to create a relationship between *'Operation'* and *'Bill of Material Relation'*, as indicated in figure 6.7. In many, more modern BIS this relationship has been implemented.

6212 What should strike in the structure in figure 6.7 is the redundancy that arises between the *'parent'* relationship type and the *'material requirement'* relationship type. The first could be dropped in an information system on condition that the *'Operation'* is recorded first followed by the required components items. But in some real life situations this is not always the case. Therefore this redundancy is normally maintained in standard BIS.



Item:

Item	Item Type	Batch size	Total Stock on Hand	Supplier
car	manufacture	1	0	-
wheel	purchase	40	49	Wheels Inc.
chassis	manufacture	1	0	-
bare chassis	manufacture	1	2	-
engine	manufacture	1	1	-
bumper	purchase	20	12	Bumpy B.V.
brake	manufacture	1	15	-
suspension	manufacture	1	3	-
paint	purchase	10.00	112.00	Paintball B.V.

Bill of Material Relation:

Parent Item	Component Item	Quantity	Creation Date
car	wheel	4	30/08/1992
car	chassis	1	12/08/1992
chassis	bare chassis	1	12/08/1992
chassis	engine	1	12/08/1992
chassis	brake	1	12/08/1992
chassis	suspension	4	07/12/1992
car	bumper	2	07/12/1992
car	paint	5.00	24/12/1992

Operation:

Item	Operation	Description	Capacity Unit	Run Time
car	10	paint chassis in paint cabin	paint cabin	16.00
car	20	attach bumpers	assy emp.	12.00
car	30	fasten wheels on car	assy emp.	8.00
chasis	10	build in car engine	engine emp.	50.00
chasis	20	build in suspension system	chassis emp.	20.00
chasis	30	fasten brakes to chassis	chassis emp.	24.00

Capacity Unit:

Capacity Unit	Description	Critcal Capacity	Hourly Rate
paint cabin	cabin with paint robot	yes	140.00
assy emp.	car assembly employees	no	81.50
engine emp.	engine building employees	yes	117.00
chassis emp.	chassis assembly	no	81.50

Availability (bucketed):

Capacity Unit	Week 1	Week 2	Week 3	Week 4
paint cabin	6x24 uur	7 x 24 uur	7 x 24 uur	7 x 24 uur
assy emp.	32 uur	40 uur	40 uur	40 uur
engine emp.	32 uur	40 uur	40 uur	40 uur
chassis emp.	32 uur	40 uur	40 uur	40 uur

Figure 6.6 - Example of a material and capacity requirement

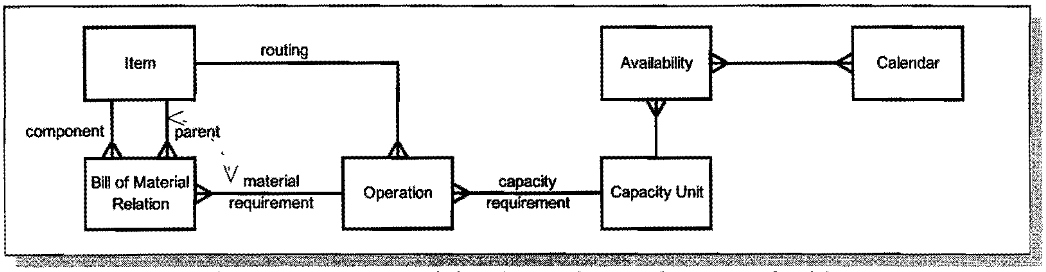


Figure 6.7 - Time-dependency in material and capacity requirements (feed-ins)

6213 Up to here the capacity requirement per '*Operation*' has only been defined singularly. If both personnel and machine requirements, and maybe other resource requirements, have to be established, in many systems this can only be solved by defining several parallel '*Operations*'. Besides an '*Operation*' with a machine requirement a second (parallel) '*Operation*' is defined with an overlap percentage containing the personnel requirement. A more fundamental solution that does not resort to this 'work-around' is realised through normalisation (see terminology in Appendix A) of a so-called n:m relationship type between '*Operation*' and '*Capacity Unit*' to the entity type '*Operation Capacity Requirement*'.

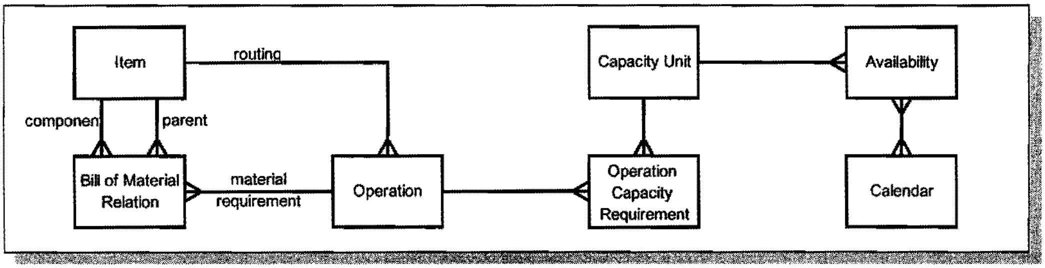


Figure 6.8 - Multiple capacity requirements per operation

6214 Though this generic elaboration may seem ideal, the various types of '*Capacity Units*' can have quite specific features of their own. At a detailed level tools, personnel, machines, ovens, silos, and other resources are quite different. A closer look is taken in paragraph 6308 and following.

6215 The next step in the development of this RDM is one that is realised in most systems, i.e. the support of multiple '*Bills of Material*' and several '*Routings*'. Five situations can cause this requirement:-

(a) *different types of 'Bills of Material' and 'Routings' for different purposes.* For engineering, it may not only be useful to have another structure, but other components may or may not be included in comparison to the '*Bill of Material*' for production. Other types are for the benefit of calculation, planning, and the definition of product variants. From paragraph 6230 on a typology is given;

(b) *alternative 'Bills of Material' and 'Routings' for the production of the same item.* If there is a shortage of components, a different product structure can be chosen, and if there is a shortage of capacity one can choose a different machine or routing. Besides a preferred bill or routing, a choice can also depend on the availability of some of these resources or even the size of a series. These selection criteria are only traceable as attribute types in the '*Bill of Material*' and '*Routing*'. Alternatives are discussed in paragraph 6234 and following;

(c) *temporary 'Bills of Material'* and more in particular those for the reprocessing of residuals, returns, rejected material, and rework. Usually, these are usually only recognised by their (short) period of applicability;

(d) *revision of 'Bills of Material' and 'Routings'* due to product or process improvements. Thus, several versions will have to be available at the same time, for example during the transition period. This applies if old components have to be used first, but also because of historical objectives. From paragraph 6249 on the subject of engineering changes will be discussed;

(e) *location dependent 'Bills of Material' and 'Routings'*, where the manufacturer has more than one production location. Despite the fact that the same product is made in several different locations, (local) differences can exist in the use of components (for example due to lower local prices) and in the capacity units present. In the discussion on 'multi-site' from paragraph 6273 further detail will be given.

6216 Apart from all these specific reasons, the recognition of multiple *'Bills of Material'* and *'Routings'* leads to an extension of the structure of the RDM with its respective entity types. The reasons for the different types are so diverse that they cannot be lumped together, though this does happen in some systems. In such cases the different *'Bills of Material'* and *'Routings'* are only numbered and the user is free to use them. The most advanced BIS make an explicit distinction of all these situations and can therefore support the corresponding advanced functionally.

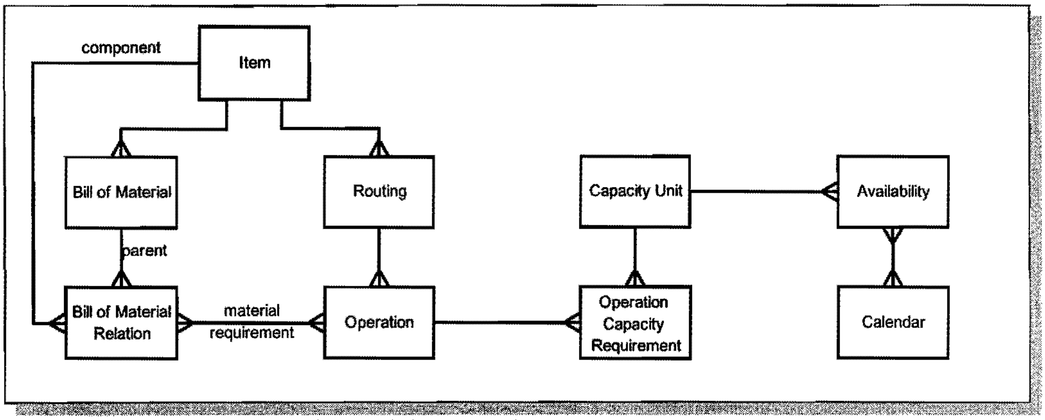


Figure 6.9 - Multiple bills of material and routings

6217 Furthermore, the possibility that a material requirement can be used in several (alternative) routings implies that the relationship type between *'Bill of Material Relation'* and *'Operation'* should be extended to a n:m relationship type[10]. In practice this is implemented by extending both entity types with a sequence number or operation number. For a *'Bill of Material'* that applies to several *'Routings'*, the restriction is that the *'Routings'* sequence numbers have to run parallel.

6218 As a last step before projecting some BIS, the entity type *'Recipe'* is introduced as it is used in various (semi-)process systems. APICS [API92] uses the following definition: „A formula or recipe is a statement of ingredient requirements. A formula or recipe may also include processing instructions and ingredient sequencing directions". In other words, a *'Recipe'* describes all that is necessary for the production of an item, i.e. the combination of required materials, capacities, additives, and instructions.

10 The normalisation of such relationship types is for the aims of the RDM, not useful see also chapter 4.

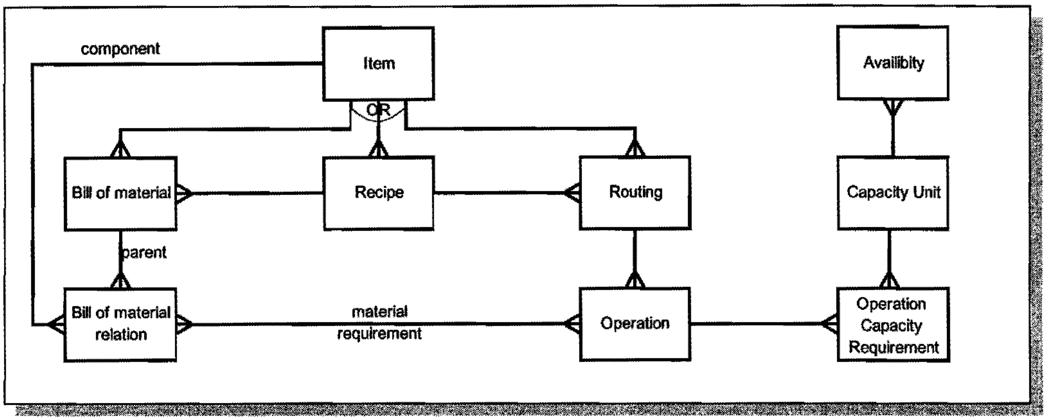


Figure 6.10 - Complex combined material and capacity requirement

6219 Due to the reasons mentioned, several *'Recipes'* for producing one *'Item'* can coexist, each with a specific material and capacity requirement. In figure 6.11 this is worked out in an example, in which a 'summer recipe' using fresh vegetables and a 'winter recipe' using frozen vegetables both leading to the same finished product.

6220 An important characteristic of this data structure is the implicit synthesis between specific structures for the (semi-)process industry and the discrete industry, with exception of the divergency aspect, which is elaborated on in paragraph 6316 and following. Note that a fundamental form of redundancy arises which is legitimate and is essential in this RDM[11], but is certainly not acceptable in a real system.

6221 Noticeable from figure 6.12 is that in the first two examples no projection is made on the *'Bill of Material'*, while nevertheless reference is made to them by projected relationship types. This means that the BIS concerned do recognise the term *'Bill of Material'*, but in this case only as an (identifying) attribute type, in the *'Bill of Material Relation'*.

6222 A brief analysis of these four sample projections:-

(a) BIS-A is a typical MRP system with traditional bills of material and routings. As is customary in these kinds of systems the capacity requirements are limited to two types (normally machines and personnel);

(b) BIS-B is a mixture of the use of bills of material and routings, as well as recipes. Here, too, the capacity requirements are limited to the usual two types. These findings suggest that the system evolved from a typical MRP system into a mixed form that can function in the more process-oriented production situations;

(c) BIS-C has a completely deviant structure, based on standard routings (these will be dealt with from paragraph 6234 on). As a consequence, no link can be made between the material and capacity requirement. The time-phased material requirement mentioned earlier can no longer be supported, making this system less suitable for production processes with long running times. However, the capacity requirement is completely normalised so that a variable number of capacity resources can be used. This combination implies for example suitability for the semi-process industry;

(d) BIS-D displays a complete decoupling between recipes (which are defined parallel to a routing). These recipes are used for completely equivalent co-products, see paragraph 6316 and following. In addition, one should see the symmetry between the capacity requirement and material requirement. This BIS treats both aspects equally.

---

[11] In chapter 4 an extensive discussion on redundancy in the RDM is given. Redundancy being necessary to compare different situations. In a real BIS, choices in the implementation of data structures have to be made.

Item:

Item	Item Type	Batch size	Supplier
spring roll	manufacture		-
fresh vegetable	purchase		greengrocery
frosted vegateble	purchase		greengrocery
meat	purchase	1,000 kilogram	butcher

Recipe:

Item	Recipe	Batch Size
spring roll	summer recipe	1,000 boxes
spring roll	winter recipe	100 boxes

Bill of Material:

BOM identification	Recipe Item	Recipe	Bill-of-Material Type	Batch Size
bill 1	spring roll	summer recipe	production	1,000 kilogram
bill 2	spring roll	summer recipe	calculation	1 kilogram
bill 3	spring roll	winter recipe	production	1,000 kilogram
bill 4	spring roll	winter recipe	calculation	1 kilogram

Bill of Material Relation:

BOM identification	Component Item (ingredient)	Quantity	
bill 1	**fresh vegetable**	**500 kilogram**	0-2
bill 1	meat	300 kilogram	0-1
bill 1	spices	10 kilogram	0-3
bill 1	**water**	**50 liter**	0-3
bill 2	fresh vegetables	0,5 kilogram	-
bill 2	meat	0,3 kilogram	-
bill 3	**frosted vegetables**	**550 kilogram**	0-5
bill 3	meat	300 kilogram	0-6
bill 3	spices	10 kilogram	0-8
bill 3	**water**	**none**	0-8
bill 4	frosted vegetables	0,55 kilogram	-
bill 4	meat	0,3 kilogram	

Routing:

Routing identification	Recipe Item	Recipe
routing 1	spring roll	summer recipe
routing 2	spring roll	winter recipe

Operation:

Operation Id	Routing identification	Operation	Setup Time	Run Time
O-1	routing 1	carving	-	1 hour
O-2	routing 1	mincing	0.15 minutes-	1 hour
O-3	routing 1	mixing	-	2 hours
O-4	routing 1	packing	1 hour	8 hours
O-5	routing 2	**defrosting**	-	**24 hours**
O-6	routing 2	carving	-	1 hour
O-7	routing 2	mincing	0.15 minutes-	1 hour
O-8	routing 2	mixing	-	2 hours
O-9	routing 2	packing	1 hour	8 hours

Figure 6.11 - Example of a recipe for a snack manufacturer

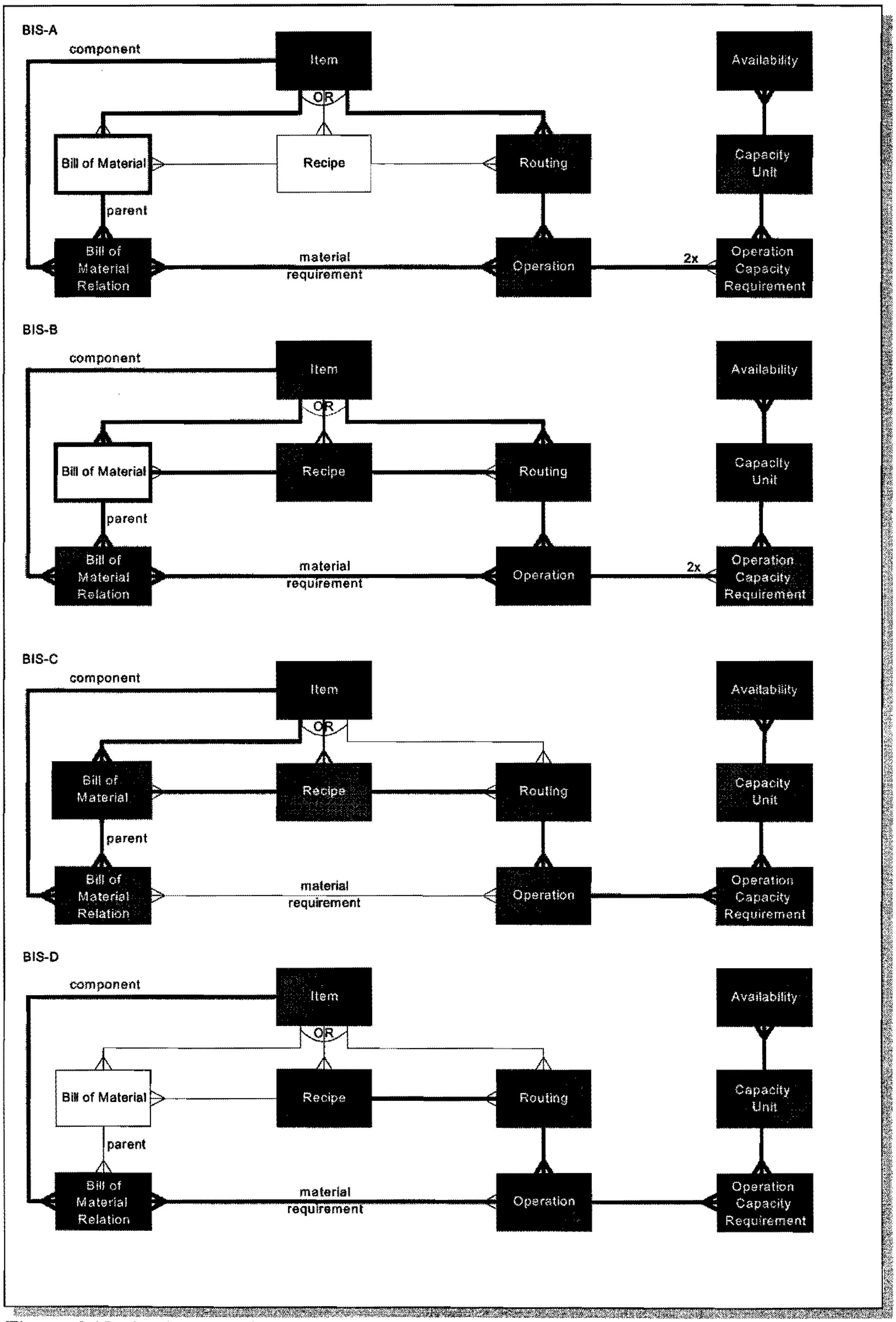


Figure 6.12 - Projection of real BIS on the RDM

6223 An important aspect that appears from the projections on this part of the RDM is the choice made in the systems between recipes, bills of material and routings. A brief comparison:

	BIS-A	BIS-B	BIS-C	BIS-D
item controlled material requirements (bills of material)	+	+/-[1]	+	-
item controlled capacity requirements (routings)	+	+/-[1]	-	-
item controlled material+capacity requirements (recipes)	-	+/-[1]	+	+

[1] Here no actual choice is made in the system, choice is postponed to the implementation

Table 6.1 - Basic comparison between the four BIS

Usage and control

6224 Here, further details are provided on the RDM, especially for using and managing 'State-independent Manufacturing Data'. The following aspects will be discussed:-

(a) *a typology of 'Items'* that are first concisely defined and expanded later;
(b) *a typology of 'Bills of Material' and 'Routings'* that are also concisely defined here, to be dealt with in more detail later in this chapter;
(c) *the standardisation of 'Bills of Material', 'Routings', and 'Operations'* due to their item independent definition, a possibly more efficient structure emerges;
(d) *the establishment of alternatives and substitutes* with which the options that are available during a material or capacity shortage are defined beforehand;
(e) *the engineering changes* where design and alteration policies are coupled;
(f) *the generic and configurable 'Bills of Material' and 'Routings'*, where a more efficient storage (and retrieval) structure is applied for situations with a large number of variations;
(g) *a classification of 'Items'* with which the retrieval and reporting of items is supported;
(h) *the elaboration for multi-site structures* in which each site has some of its own data which is defined under the overall co-ordinating data;
(i) *the registration and conversion of various units of measurement for 'Items'*.

6225 Where useful, a real BIS is projected in parts of the RDM to give an interesting illustration of existing systems. In contrast with the previous part, because of the total complexity, it is no longer possible to combine all these RDM-diagrams into one data structure.

Typology of items

6226 All BIS have some kind of item master file in which usually many hundreds of attribute types are stored. This complexity is mainly caused by the diversity of items in use, their management and the resulting characteristics. In figure 6.13 this variation is normalised in the form of a data structure. An attempt is made to include all the important and most customary specialisations in this structure. Finally, some very special entity types are mentioned (drawing, tools and power) that are considered to be items in some BIS, but differ to such an extent that in this context they are treated separately.

6227 Following the structure of figure 6.13, a further explanation of these specialisations of the entity type '*Item*' is given:-

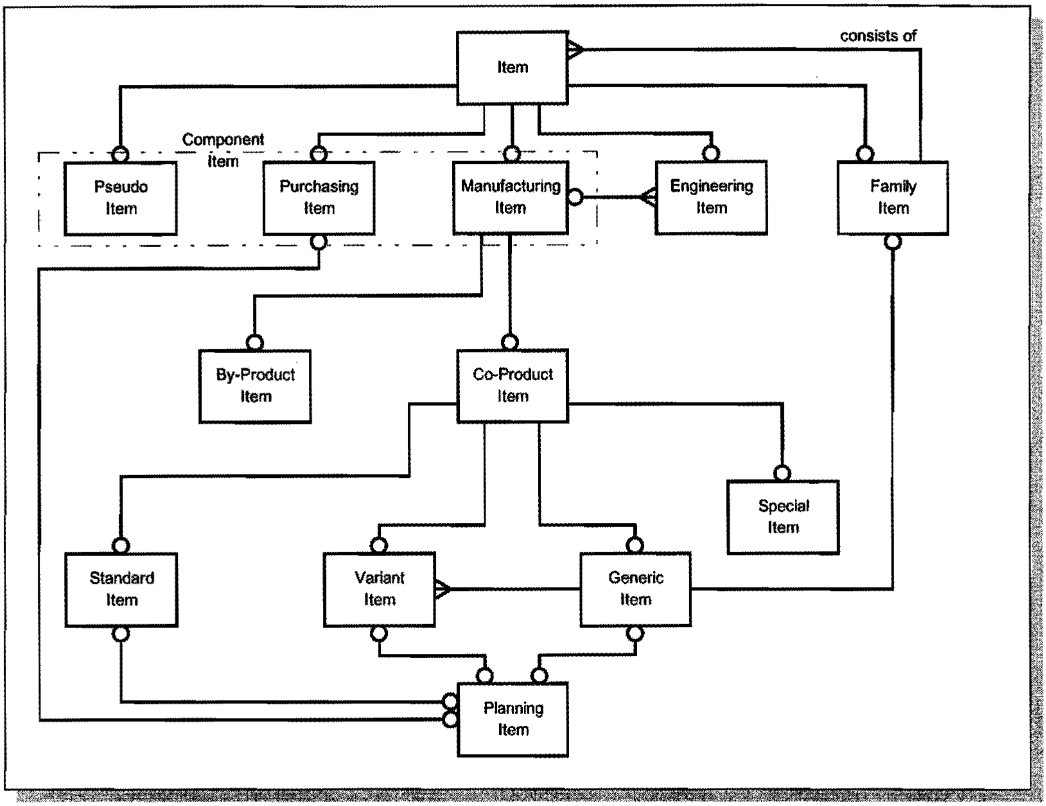


Figure 6.13 - Item typology

(a) *pseudo items.* These are items that are not physically exist, but are used for structuring a '*Bill of Material*'. Using a pseudo as a sort of dummy, an extra level can be made in a '*Bill of Material*' with which a collection of components is grouped together. This is mainly applied to support line production and to be able to define modular material structures. Some essential characteristics of '*Pseudo Items*' (particularly for the MRP calculation) are: zero lead time, non-stock and lot-for-lot replenishment;

(b) *purchase items.* These are all items that can be purchased, sometimes depending on the production capacity which means that this type is not distinct from the '*Manufacturing Item*'. In chapter 5 where the RDM for 'Purchasing' is described, this type is detailed. The relation between production and purchasing, and more especially the case of subcontracting, is dealt with in paragraph 6281 and following;

(c) *manufacturing items.* These are all semi-finished products and finished products produced by the enterprise, including production by means of subcontracting. Here, a distinction can be made between:-

(i) *by-product items.* These are all important products (based on value or environmental requirements) that are an unwanted result of or remain after the production process. The balance between '*By-product Items*' with respect to primary products is often known in advance and is recorded in the '*Bill of Material*'. Where possible '*By-product Items*' are recycled, reused, sold or disposed of at cost (where this happens within the same production process, it creates a cyclical loop in the material requirement structure);

(ii) *co-products.* These are all primary products that are the required (planned) result of the production process. The prefix 'co-' suggests situations in which more than one product is produced, as in material separation processes. These products are completely equal by definition, however, the term co-product can also be used for normal singularly produced manufacturing items. To make the distinction between co- and by-products clearer, it can be stated that a co-product will be controlled by a master production schedule, or be produced to customer order.

A further elaboration of co- and by-products in data structures is given in paragraph 6316 and following.

In relation to the (customer) specification of the manufacturing item, following distinctions can be made:-

(iii) *standard items.* These are items that are sold to different customers and/or are used in several finished products. In practice this standardisation often justifies stock keeping to some extent;

(iv) *generic items.* These are items where instead of one or more fixed components choice is made from ranges or enumerations of optional components. Due to the large number of possible combinations such items are not defined as separate *'Items'*, but only as a selection structure. The definition of a *'Generic Item'* can be based on product features or characteristics, which are translated into components by means of questions and answers. From paragraph 6257 onwards 'product generators' will be discussed;

(v) *variant items.* These items are the result of a configuration process of generic items as described above;

(vi) *special items.* These items are completely customer-specific, they are designed and specified for a particular customer. These customer-specific items are dealt with from paragraph 6289 on, the relation between production and sales is discussed;

(d) *engineering items.* These are items that are the subject of the design process. These need not necessarily be customer-specific finished products (like *'Special Items'*), but could also be standard finished products or semi-finished products instead. Engineering changes are discussed in detail from paragraph 6249 onwards;

(e) *family items.* These are groups of items to be treated equally, due to similarity of design, the production characteristics, or the pattern of purchasing. *'Family Items'* are often used for forecasting and planning purposes (see *'Planning Bills'* from paragraph 6230 onwards) or for classification (see paragraph 6269 and following). In essence the objective of family items is comparable to that of generic items, only the structure is much simpler and for example contains no features and options;

(f) *planning items.* These include all the previously mentioned items where stock and production is planned and controlled. Besides the related items in figure 6.13, it can also be useful at the level of 'master production scheduling' level to define capacity units as planning items. This is worked out in paragraph 6230 and following in the section about *'Planning Bills'*.

6228 Besides this definition, a distinction is also made between raw material, auxiliary material, semi-finished products, and finished products. In principle this distinction can easily be traced in the typology above and requires no further elaboration. In a number of BIS some additional resources are treated as an item, nevertheless they have such different characteristics compared to materials that they are discussed separately in the RDM. It concerns:-

(a) *power.* It is usually available in unlimited amounts, but for certain combinations of production operations there could be a temporary overload. Though there may seem to be a material usage (i.e. energy) the planning characteristics are more comparable with those of capacity, since after an operation sufficient power is again available. This resource is treated in more detail from paragraph 6308 onwards;
(b) *drawings.* These are related to the design process and thus have a relationship with the *'Engineering Items'*. *'Drawings'* are not consumable for example, and if a *'Drawing'* has been made we can make as many copies as are necessary. For more detail see the engineering documents from paragraph 6249 onwards;
(c) *tooling.* Though tools resemble capacity resources, there are situations in which tools have to be produced or purchased. Another phenomenon that has material-like characteristics is the wear of some tools. Nevertheless, the establishment, constraints, and allocation mechanisms are more related to capacity resources and are treated as such from paragraph 6208 onwards.

6229 The problem is to recognise the above item entity types as a specialisation of the 'item master file' in a BIS. This is only possible through a detailed study of the (hundreds of) attribute types. A better method is to deduce this from the relationships between *'Item'* and other entity types. In the following paragraphs as much use as possible of these specialisations is made, so that conclusions can be drawn about their presence in illustrative projections of BIS.

Typology of material and capacity requirements

6230 As indicated earlier various types of material and capacity requirements (*'Bill Types'*) can be defined. A generic structure is shown in figure 6.14. Various BIS have chosen for this solution.

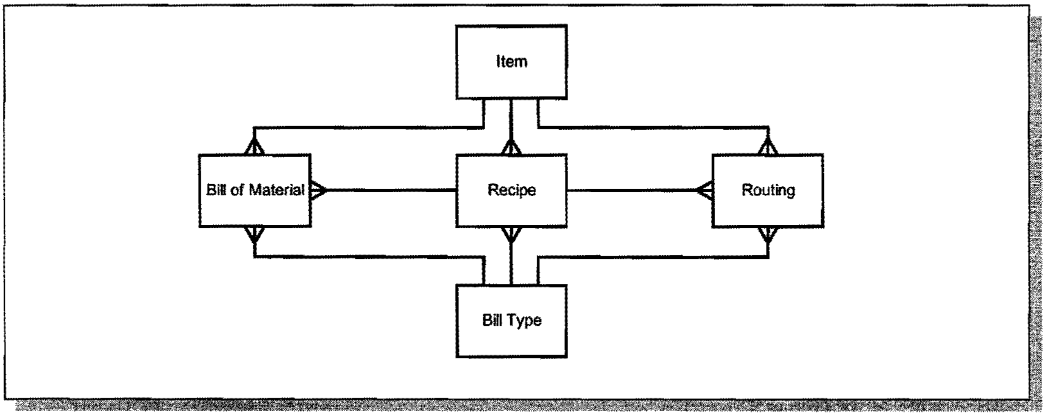


Figure 6.14 - Generic implementation of a bill typology

6231 Whatever structure is implemented, the required functionality is far more extensive than follows from the addition of a new type in this generic data structure. So from the presence of an entity type *'Bill Type'*, only very few conclusions can be drawn. Here the most important types that can be distinguished are discussed:-
(a) *manufacturing bill.* This is essential in every BIS and is normally implied when there referring to a *'Bill of Material'*, *'Recipe'*, and *'Routing'*;

(b) *costing bill.* This can be considered to be a simplification of the *'Manufacturing Bill'* that is used for the precalculation of the cost price. So this *'Bill'* is not essential in cases where a cost price of zero can be linked to the irrelevant parts of the *'Manufacturing Bill'*. As soon as matters such as the guarantee, packaging material, and documentation play a role, which are not part of the *'Manufacturing Bill'*, a separate entity type is required. In such more complex situations, the separate specification of a *'Costing Bill'* and the ability to perform simulations can be useful;

(c) *sales bill.* This is used for the sale of composite *'Items'* that are only picked and shipped together, basically as a kit. An example is the sale of Christmas hampers, in which all the components are allocated separately in the warehouse and are only packed together for shipment (the idea is that the packaging is so simple that this does not need to be managed as a process). By definition a *'Sales Bill'* only involves a material requirement and not a capacity requirement. Though some BIS have a separate *'Sales Bill of Material'*, this type can also be implemented in the general *'Manufacturing Bill'*. Then a distinction is made by means of the *'Bill Type'* or by a (controlling) item attribute;

(d) *purchasing bill.* This is used in special situations where it is necessary to know the components used by the supplier during the production of a purchase item. For example, where repairs are to be carried out in-house. For this type of bill only the material structure is relevant, and an obvious choice is to implement it as a simpler version of the *'Manufacturing Bill'*, by means of a *'Bill Type'* or by an item attribute;

(e) *subcontracting bill.* This structure is a mixture between the *'Manufacturing Bill'* and the *'Purchasing Bill'* and among other things it defines the materials required by the subcontractor, next to the other in-house *'Operations'* to be carried out with their material requirements. This type is always combined with the normal *'Manufacturing Bill'*. The particulars are dealt with from paragraph 6283 on;

(f) *engineering bill.* This describes the material and processing structure that are part of and the result of the design process. This implies that the components of this bill type are equivalent to the final *'Manufacturing Bill'*, but that it has a different hierarchical structure. All BIS giving extensive support to the engineering process recognise this type as a separate data structure. From paragraph 6249 onwards the re-engineering process and the specific *'Engineering Bill'* are dealt with;

(g) *generic bill.* This describes a material, processing, and calculation structure based on optional components or features with product characteristics. The more advanced forms are so different from the *'Manufacturing Bill'* that they are implemented as a separate data structure. This is discussed in paragraph 6257 and following;

(h) *configured bill.* This describes the specific result of the configuration process on a *'Generic Bill'*. This bill is usually recorded in the standard *'Manufacturing Bill'*, because of the similarity in structure, or it is directly stored in the customer order related production order. For more details see paragraph 6257 and following;

(i) *planning bill.* This only describes the *'Items'*, *'Capacity Units'*, and other resources that are critical to the planning process. Nevertheless, the complexity can vary from a simplified version of the *'Manufacturing Bill'* to a complex structure in which the parent and the components can be *'Generic Items'* (see [VEE91] and [HEG95]). This complexity does not become visible in figure 6.15, but is obvious in figure 6.23. For the *'Planning Items'* prognoses are made and forecast consumption takes place at the acceptance of customer orders. The *'Item'* ordered is related to a *'Family Item'* by means of a *'Planning Bill'* or this *'Item'* has a more complex relationship with a *'Generic Item'*. It is at this higher level that the forecast consumption takes place. Note that a *'Planning Item'* not only refers to material, but should also be capable of controlling critical *'Capacity Units'*. A *'Bill of Capacity'* or *'Bill of Resources'* as defined in the APICS terminology [API92] for performing a Rough Cut Capacity Planning (RCCP) is therefore a form of *'Planning Bill'* as defined here.

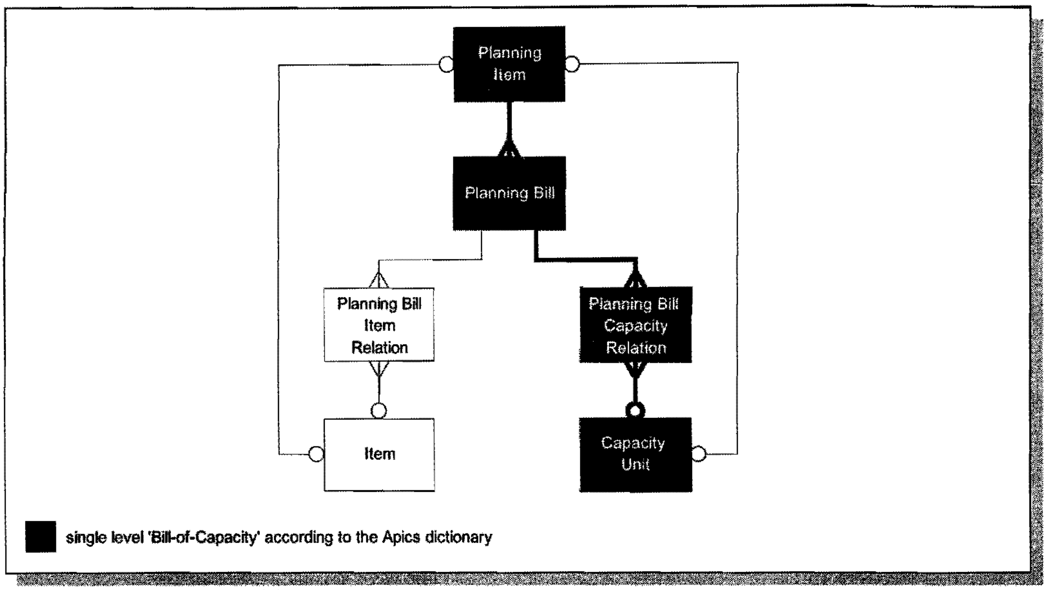


Figure 6.15 - Planning bill

6232 In a number of BIS no separate *'Planning Bill'* can be distinguished, but a hard coded part (for example only material) of the *'Manufacturing Bill'* is used for this purpose. Generally speaking these are the BIS with minimum support for master production scheduling.

6233 Summarising, often the following (separate) entity types in the data structures of existing BIS can be distinguished: *'Manufacturing Bill'*, *'Engineering Bill'* for the specific support of the engineering process, *'Generic Bill'*, and *'Planning Bill'*. *From this we can conclude that the recognition of typologies in existing BIS data structures is only possible to a limited extent. Modelling of such a typology is possible, as can be found in the RDM and is clearly demonstrated in figure 6.13. This means that if the typology supported by an existing BIS has to be investigated (once) because of the projection, then the projection on the RDM is a good medium for storing the results.*

Standardised material and capacity requirements

6234 In more process-oriented industries, it appears that the material structure is very simple and even is identical for a large number of finished products. In that case the variety of finished products is determined by the production process. One trivial example is the grinding of stone into talc, the variation in particle-size is determined by the *'Operation'*. Another example is the production of different kinds of alcohol. The reverse also occurs, i.e. a standard process with which different finished products can be made depending on the ingredients.

6235 Here, the focus lies on material and process structures that are defined independently of the *'Item'* to be produced and which are combined in an item-specific recipe[12]. A number of different standardised entity types can be distinguished:-

12 Note that there is a difference between standard and referential entity types. 'Standard' means that the same data is used unchanged (and even can not be changed) in different situations (for example items). 'Reference' means that data is copied to a new structure and can be changed as required. 'Reference' is only useful when making new structures.

(a) *'Standard Bills of Material'*, in which a material structure is recorded independently of the *'Item'* to be produced, instead the finished product is determined by the production process. Though there are various industries in which this is the case, this is hardly ever supported by the existing BIS. A simple 'work-around' is based on a *'Pseudo Item'* for which the standard *'Bill of Material'* is recorded. In every item-specific *'Bill of Material'*, only the requirement for this *'Pseudo Item'* is recorded;

(b) *'Standard Routings'*, in which a process structure is recorded independently of the *'Items'* to be produced. This structure occurs in BIS that focus on semi-process industries;

(c) *'Standard Operations'*, in which individual operations are recorded as standard and where it makes little difference what this operation is released on. One example is the sawing of pipe material, where the length and diameter of the *'Item'* is not relevant, as same sawing time and machine are involved in all cases.

6236 The *'Standard Bills of Material'* and *'Standard Routings'* are used to define a *'Recipe'* for an *'Item'*. In figure 6.16 this structure is shown. Note that because of the combination of concepts (standard versus item-specific structures) there is much redundancy in the RDM, which would be most undesirable in a real implementation[13].

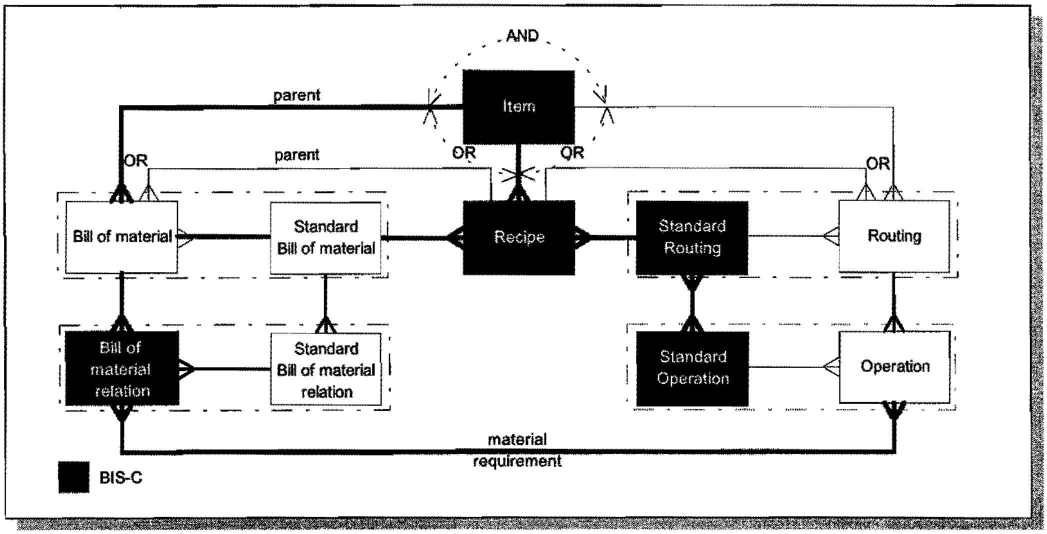


Figure 6.16 - Combined standard and item-specific and operating requirements

6237 The BIS projected in figure 6.16 shows a redundancy relationship between *'Item'* and *'Recipe'* (i.e. directly and via the *'Bill of Material')*. This is due to being able to define a material requirement independently from a *'Recipe'*. Furthermore, *'Standard Operations'* can be established. It can be deduced from the n:m relationship shown that these *'Standard Operations'* can be related to several *'Standard Routings'*.

6238 The use of standard entity types diverges strongly from the concept in the RDM in figure 6.10 and therefore from the traditional MRP structure. Combining both concepts in one BIS leads to (undesirable) complexity. In the RDM this combination gives us interesting insight into the concepts in different BIS, see for example the projection in figure 6.16. Since combinations of these concepts in BIS are highly improbable, the RDM can be simplified a little (see figure 6.17).

13 This means that a BIS will not cover this combination and therefore can not be optimally suitable for every situation.

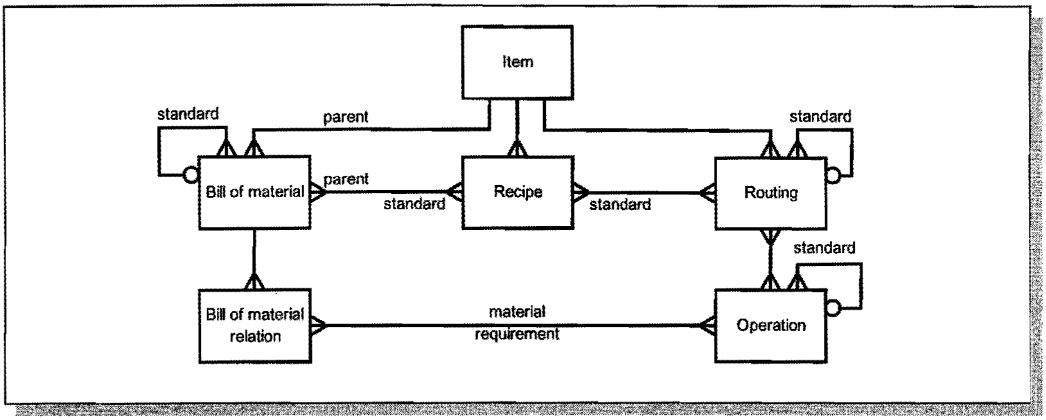


Figure 6.17 - Simplified model for standard and item-specific and operation requirements

6239 If a BIS does not support the use of standard structures, there are a number of 'work-arounds' possible to create some standards. As mentioned earlier '*Pseudo Items*' can be used to simulate a '*Standard Bill of Material*', so can '*Standard Routings*' be defined for a '*Pseudo Item*'. They function as a reference and have to be copied to the particular '*Items*'. Some disadvantages are the extra storage space required and the management of changes.

6240 On the other hand, if a BIS only supports standard entity types, while item-specific structures are required, the only solution is to define separate '*Standard Routings*' and '*Standard Bill of Material*' for each '*Item*'.

Alternatives

6241 Depending on the availability of material and capacity, one should be able to resort to alternatives. Basically there are two kinds of alternatives or substitutes that can be distinguished:-

(a) *fully equal alternatives*, where the preference only depends on the circumstances. An alternative is chosen based on availability, capacity utilisation, or lot-size. Selection is then a complex form of decision support and falls under detailed planning. Basically, the requirements are defined at a higher (group) level, for example at the machine group level rather than individual machines. In such a case the capacity unit in the RDM is this higher level. This type of decision support and the corresponding data structure are dealt with from paragraph 6295 onwards;

(b) *unequal alternatives*, where an absolute preference exists, independent from the circumstances. The alternative is only used when unplanned shortages occur. These are treated as exceptions that in principle are signalled by the BIS, including an indication of the next preferred alternative. Adaptation is usually done manually and only requires limited support.

6242 The next paragraphs discuss the *unequal alternatives*. Further it should be noted that such an alternative can be defined for nearly every entity type in the RDM. In this context alternatives are described for the most relevant entity types. In figure 6.18 the alternatives for the material requirement structures are shown.

6243 In figure 6.18 the '*Alternative Items*' are modelled in such a way that they can be valid only in the context of a particular '*Bill of Material Relation*'. In such an alternative additional attribute types can be defined such as the '*degree of preference*' (usually including a sequence number), a '*deviating amount*' or a '*conversion factor*'.

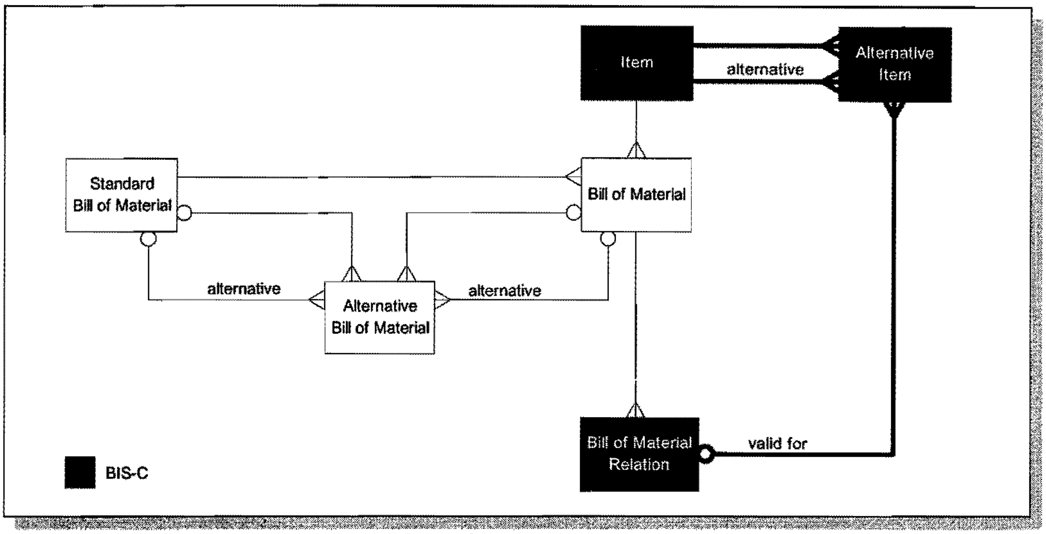


Figure 6.18 - Alternatives in material requirement

6244 *'Alternative Bills of Material'* are also modelled in figure 6.18. In BIS such a structure does not occur often. Instead, the choice is often made to record the *'Alternative Bill of Material'* and the regular *'Bill of Material'* in one entity type (see also paragraph 6215(b)). The distinction is only made by determining the *'degree of preference'* for each of these bills. This simplified structure would seem to offer sufficient functionality in almost every situation. This is not the case if this structure were to be chosen for *'Alternative Items'*. Here, normally need arises for more complex replacement structures, like the following hierarchical structure: C is an alternative for A and B; and B is an alternative for A.

6245 As for *'Alternative Bills of Material'*, *'Alternative Recipes'* can be defined, see figure 6.19. Here, too, alternatives are always distinguished by a sequence number in the *'Recipe'*, indicating the *'degree of preference'*.

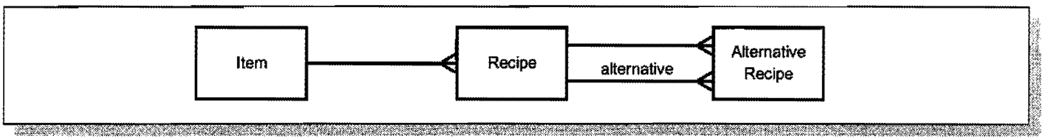


Figure 6.19 - Alternative recipies

6246 In figure 6.20 alternative operation structures and capacity requirements are modelled in a comparable way. Here, also *'Alternative Routings'* are almost always distinguished by a sequence number in the *'Routing'*.

6247 The provision of *'Alternative Capacity Units'* occurs in most BIS by means of alternative operations. If another machine is chosen this usually influences the operation time and other operation-related characteristics.

6248 In short, *'Alternative Items'* even context-dependent *'Items'* and *'Alternative Operations'* can be traced directly in the BIS data structures. Support for alternatives of the other entity types mentioned has to be sought for in attribute types such as *'degree of preference'* or *'sequence number'*, which usually is not that difficult.

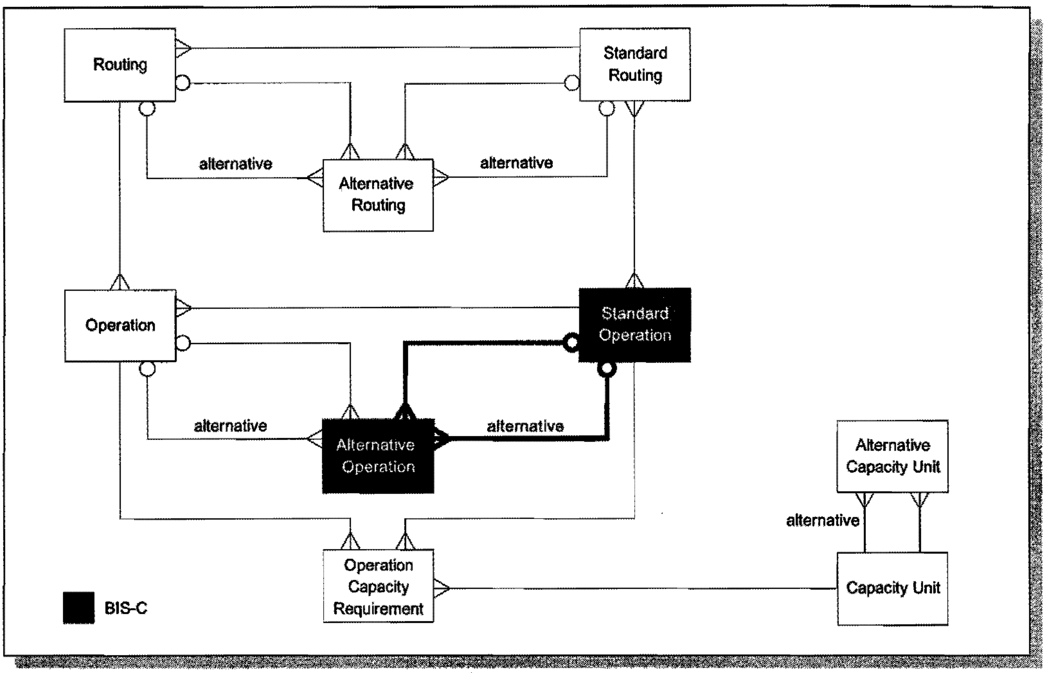


Figure 6.20 - Alternatives in capacity requirements

Engineering

6249 The functionality concerning the engineering process shifts quickly to the area of Computer Aided Design (CAD). The interface or integration between CAD systems and BIS is becoming increasingly important, especially because of the growing role of total quality systems and the trend towards greater customer-specialisation. As far as BIS support is concerned, engineering items, engineering documents, and engineering changes are discussed here.

6250 In the typology of *'Items'* (paragraph 6226 and following), *'Engineering Items'* were already introduced as a specialisation of this entity type. As version management is particular important here, the entity type *'Engineering Item Revision'* can also be distinguished. This structure is shown on the left-hand side of figure 6.21, as well as the relationship with the specific material structure, namely the *'Engineering Bill of Material'*.

6251 In the same figure the *'Engineering Documents'* (sometimes incorrectly put on a par with drawings) are also shown and is extended with an entity type for version management. It should be noted here that these documents are not only related to the product design, but also to the production process design. The relationship between *'Engineering Bills of Material'*, *'Recipes'*, and *'Routings'* is established via the *'Engineering Items'* in the RDM (see figure 6.22). By means of a recursive relationship to itself, complex *'Engineering Document'* structures can be defined. Furthermore, reference is made to these documents by *'Item'* and *'Engineering Item Revision'*, including, if possible their positions on the drawings in question.

6252 In most BIS the *'Engineering Documents'* are not normalised into separate entity types. Instead, the *'Engineering Item'* refers to a drawing. In the projected BIS such a reference is also made, which also establishes the positions on the drawing.

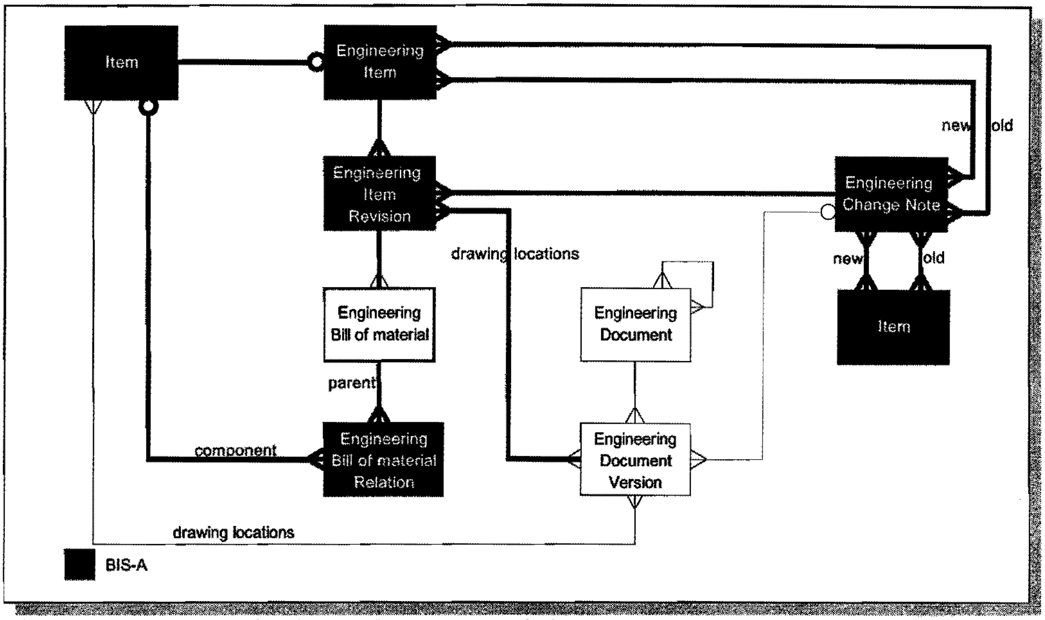


Figure 6.21 - Engineering items, documents and change notes

6253 Finally, on the right-hand side of figure 6.21, '*Engineering Change Notes*' are present for the management of changes on existing material and operation structures. Of course a relationship exists between the '*Engineering Change Notes*' and the '*Engineering Documents*' that may be the source of the changes. The relationship types between the '*Engineering Change Notes*' and other relevant entity types are shown in figure 6.22.

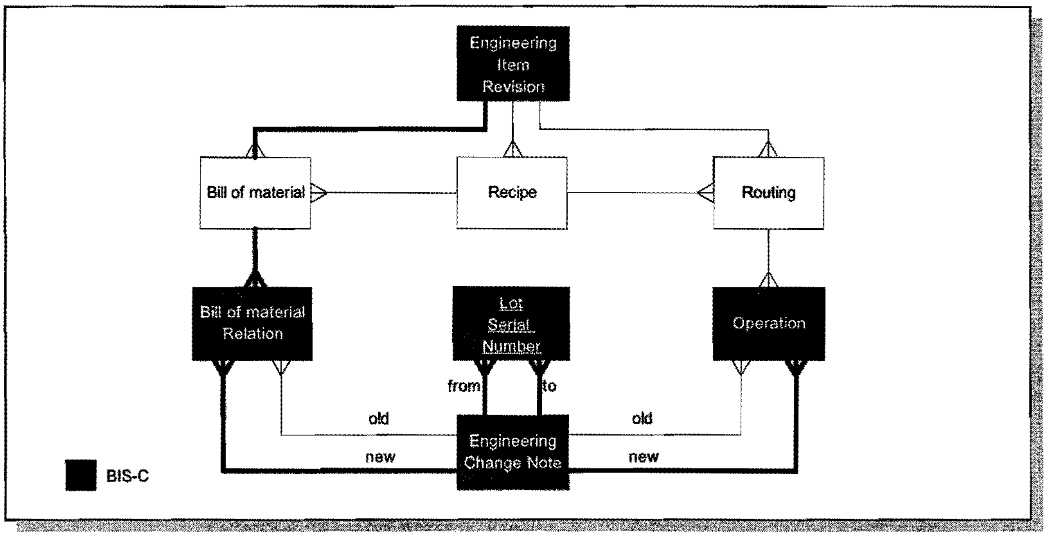


Figure 6.22 - Engineering changes

6254 In various BIS validity dates can be given for entity types such as '*Bill of Material Relation*'. By means of '*Engineering Change Notes*', as in figure 6.22, such changes can be recorded in relation to each other. Moreover, it is possible to retain more details about changes, such as a reason for change, an owner, and the status.

6255 Regarding the execution of engineering changes, three forms can be distinguished:-

(a) *time dependent.* This is by far the simplest form and is based on a previously determined starting date on which the change takes effect. This form is recognised by all BIS supporting engineering changes;

(b) *stock dependent.* This form can be subdivided into:-

  (i) executing the change when the stock of old components is completely exhausted, this prevents obsolete stocks;

  (ii) executing the change when all new components are available, so production is not hindered by a shortage of some components.

  Both these forms are supported only by a few BIS and they are not directly visible from the data structure. Such complex logic, for example in the material requirement calculation, will only be based on an indicator in the entity type '*Engineering Change Note*';

(c) *serial number or lot dependent.* Here, changes are carried out from a certain serial number (or lot number) onwards. This special form occurs, for example, in the car manufacturing where a large number of changes are synchronised by executing them from a certain serial number onwards. This is also the time at which the revised version of the technical documentation (for the benefit of dealers) becomes valid. This form can only be found in specialised BIS. In the RDM this relationship has been included, and is also supported by the projected BIS (i.e. used in a number of car industries!).

6256 If engineering is an important aspect of business management, besides the functionalities described here, attention will have to be paid to the degree of integration with specific CAD systems.

Generic and configured bills

6257 As a result of the enormous increase in the product variation, more and more BIS support the generic definition of items and their underlying structure. This leads to very extensive functions comparable to an expert system. It would be tedious to investigate such complex rules based on data structures alone. In this context the RDM is limited to a general model depicting only the essential entity types for the support of simpler and more complex structures. For an extensive elaboration the reader is referred to [VEE91] and [HEG95]. Three areas are distinguished here:-

(a) *the definition* of generic material and operation structures;

(b) *the specification* of a variant based on such a generic structure;

(c) *the storage* of the variant resulting from the generation / configuration process.

6258 The definition of generic material and operation structures takes place using features and options. Features are related to the mandatory choice from a collection of components, while options are defined as the components from which there is a facultative choice (see definitions [API92])[14]. Thus, the sunshine roof in a car is a facultative option, while the engine is an mandatory feature. For that matter, more complex relationships between components can occur, which leads to separate entity types for inclusion and exclusion of complex component relationships. With this the need for the restricted distinction between features and options disappears. Besides, the choice for or against certain features and options may have consequences for the operations (and of course for the cost price calculation). Figure 6.23 shows the general structure of the RDM this general nature stems from the various n:m relationship types.

---

14 The more advanced product configurators do not use the terms 'features' and 'options' anymore but 'parameters'. In this generalisation a parameter refers not only to components, but can also describe properties or characteristics of items. In the data structure these terms are treated as equal.

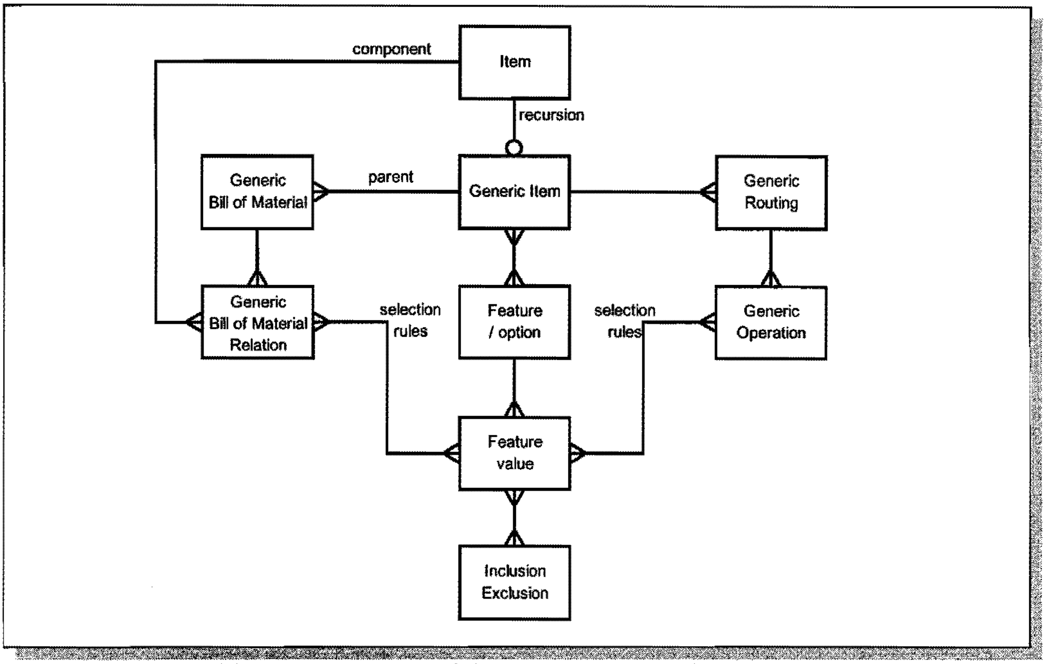


Figure 6.23 - Definition of generic material and operation requirements

6259 In a number of BIS the features and options can only be defined at the highest level of the generic structure and a *'Generic Item'* cannot be a component in a *'Generic Bill of Material'*. In figure 6.23 this will only be clear by the omission of a relationship type between *'Item'* and *'Generic Item'*.

6260 This recursive relationship (between *'Item'* and *'Generic Item')* is one of the essential characteristics of Van Veen's work [VEE91]. Other characteristics of this general data structure are:-

(a) the selection mechanism from the *'Generic Bill of Material Relations'* (and *'Generic Operations')*;
(b) the obligation to define the configuration parameters at the highest level[15];
(c) the consideration that the sales and manufacturing structure have to be identical.

6261 The structure of *'Generic Bills'* seems simple, but much of the complexity is hidden in the *'Selection Rules'* and the *'Inclusion/Exclusion'*. The simplest implementations of this generic structure are to so-called modular structures[16], see figure 6.24.

6262 Important restrictions of modular structures are (see also [VEE91]):-

(a) every module belongs to exactly one option and the modules are completely hierarchically defined. In practice, however, the selection of components depends on a *combination* of options;
(b) the structure of the actual *'Manufacturing Bill of Material'* is lost because it is subordinate to the ability to model the *'Generic Bill of Material'*.

15 Based on the assumption of direct identification. Hegge [HEG95] also introduces indirect identification of items. Such parameter driven indirect identification is not yet available in BIS (except for a prototype developed with Baan).

16 Modular structures are connected to planning modules. A module consists of a collection of components to produce a finished product or a product family with a certain option (or combination of options), as well as a collection of components used in every finished product of this product family (see also [VEE91] and [ORL72]).

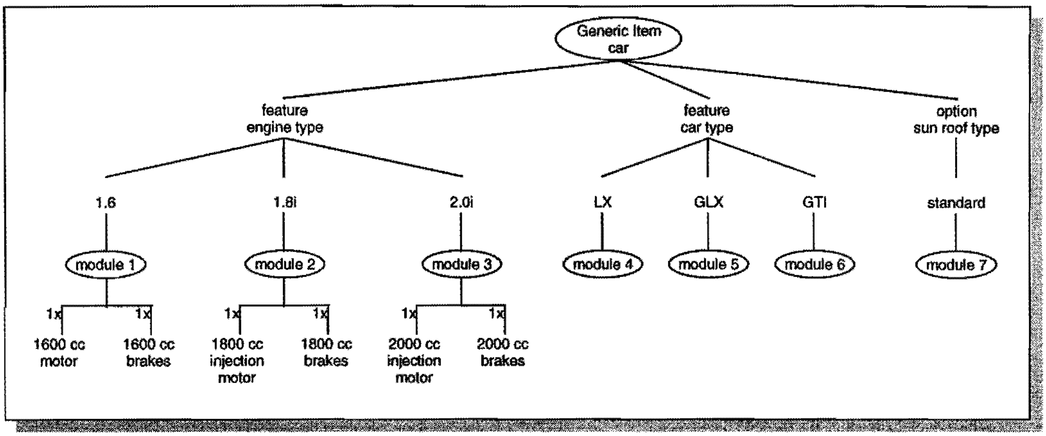


Figure 6.24 - Example of a modular bill of material

6263 In the more enhanced implementations, the '*Selection Rules*' contain very complex structures, so-called 'rule bases' with which the restrictions mentioned can be compensated for.

6264 A variant based on such a generic structure is specified by means of questions and answers. This can be done quite easily, i.e. by going through all features and options and making a choice. However , the more advanced systems are based on the following:-

(a) a formulation of questions (and answers) that are more concerned with the application than the technical structure. The answers then have to be translated into (a combination) of features and options;

(b) a question structure in which all superfluous questions are automatically skipped, or in which the next question depends on the answer to the previous question. This is clarified in figure 6.25.

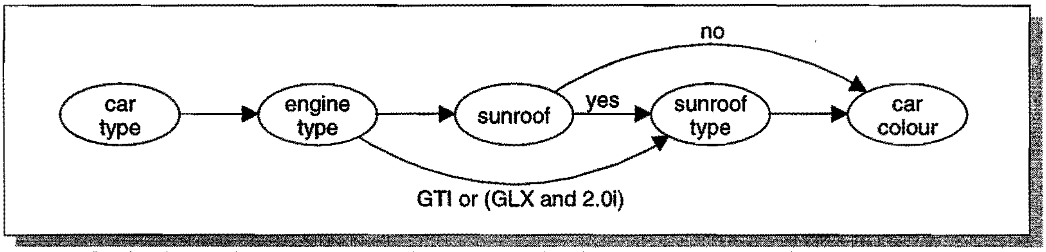


Figure 6.25 - Example of a structured questionnaire for features and options

6265 Figure 6.26 shows the data structure in the RDM of the questions and answers. In the simples model mentioned before, the relationship types between '*Question*' and '*Feature*', and between '*Answer*' and '*Feature value*' are always defined as 1:1.

6266 Finally, the variants resulting from the generation or configuration process have to be stored. This will always be in a production order (or project order), in which all configured operations and materials are included. However, for product history, reuse for the same customer or other customers, or repairs, the details need to be recorded longer than the limited life span of the production order. There are two methods for storage:-

(a) the registration of all parameters (in fact the questions and answers) used in the configuration process, so that the process can automatically rerun to determine the variant. This method has some disadvantages:-

(i) the '*Generic Bill*' has to be kept in total. After engineering changes, older versions have to be preserved in the same data structure. Only then is a rerun of the original configuration process possible. On the other hand, an advantage is that the configuration process can be rerun on the basis of the new '*Generic Bill*';

(ii) every time the variant is required it must be completely regenerated. This usually takes place automatically (no manual input for the questions and answers), but there is some delay before the variant is available;

(b) the registration of the resulting material and capacity structure as a regular '*Bill of Material*' and '*Routing*'. This version of the '*Manufacturing Bill*' is also known as the '*Configured Bill*'. This method also has some disadvantages:-

(i) storing all the configured variants means using a large database with the risk of pollution by older no longer relevant variants. The objective of '*Generic Bill*' was to reduce the large number of individual bills stored;

(ii) all engineering changes have to be carried out (manually if necessary) in these separate structures too.

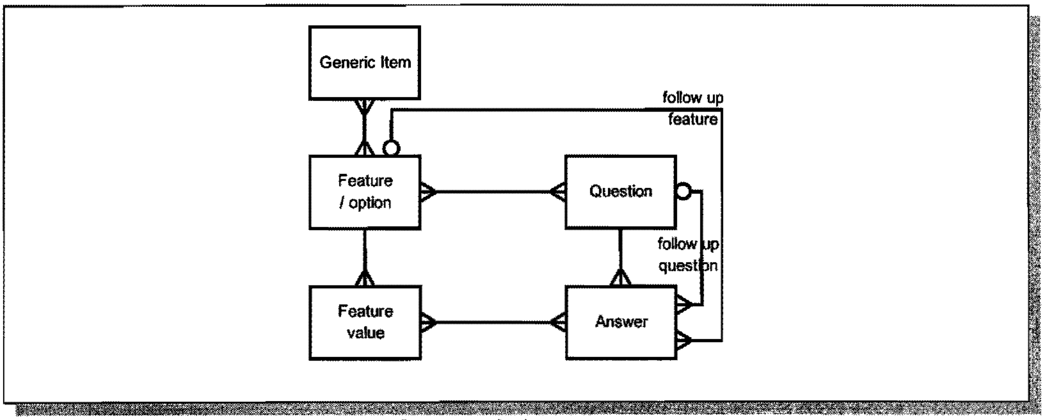


Figure 6.26 - Questioning structure for generic items

6267 Naturally, a combination of both is possible too. The RDM in figure 6.27 shows the first method mentioned.

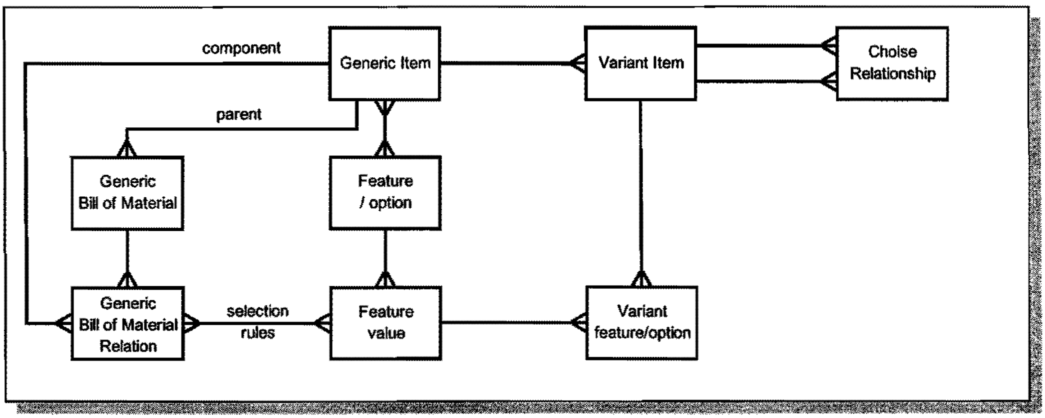


Figure 6.27 - Parameter structure for variant items

6268 For the second method additions to the RDM are not necessary, as this has already been catered for as a separate '*Bill Type*' (see paragraph 6230 and following).

6269 An item classification system is essential for large volumes, many companies have tens of thousands or even hundreds of thousands of item numbers. The classification is necessary on the one hand for retrieving items and on the other for grouping and reporting. Traditionally, some classification was used in a meaningful item coding. However, the disadvantages prove to be enormous:-

(a) the chosen coding structure will be full and there are no more codes available;
(b) the structure is only hierarchical by definition and each related coding group is a subgroup;
(c) only one structure can be used, so it has to be perfect (which leads to endless discussions and extremely complex coding structures);
(d) a changed item classification leads to a change in its unique identification.

6270 A simple option that is offered in nearly all BIS, is searching by means of an aggregation of items: family item or item group. Often searching or reporting on several of such independent item groups is possible. In figure 6.28 this is shown by the relationship type between '*Family Item*' and '*Item*'. By considering the '*Family Item*' as a specialisation of '*Item*', or by relating a '*Family Item*' directly to itself, two query structures are created. Nevertheless these relatively simple structures are not yet supported by many BIS.

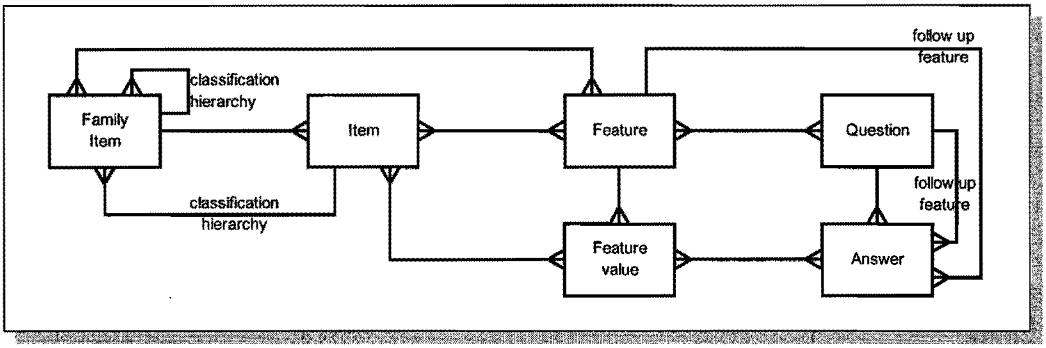


Figure 6.28 - Classification structure

6271 A more advanced classification structure is based on '*Features*' of '*Items*' or '*Family Items*', which identifies their characteristics. '*Features*' such as colour, diameter, shape, and material type can be recorded with their actual value ('*Feature value*'). This structure can also be extended with a hierarchy of '*Features*' and '*Questions*' with which a feature such as 'diameter', only applies to items where the feature 'shape' has the value 'round'. This hierarchy is modelled by means of '*follow-up features*' and '*follow-up questions*'. This structure is nearly identical to the question and answer structure in the configuration of generic bills. In BIS the obvious choice is to combine the two.

6272 The question and answer structure can be extended to an entity type complex with combinations of '*Answers*' and '*Feature values*' which are or are not permissible. Instead of such a complex data structure, this logic can be defined by the users using a logical language (for example Prolog). Clearly the further elaboration of such complex data structures for tracing further details is beyond the objective of this research.

Multi-site

6273 Finally, multi-site structures in which plants or locations, summarised under the heading '*Site*', are discussed and added to the RDM as an entity type. The entity type '*Site*' has a number of its own attribute types, for example: address data. A large number of control parameters are usually also defined at this level. However, the most important aspect is its role as an identifying attribute type for many other entity types that can therefore be made site-specific, for example: different routings per site. Theoretically, each entity type in the RDM can be extended in this way, but there are some practical limitations.

6274 Figure 6.29 shows the multi-site structure in the RDM. Here an extra entity type has been added in between '*Site*' and '*Capacity Unit*'. For in many production organisations there is a higher level than the capacity unit on which production is directly planned. This higher level, usually called a production unit, department or cost centre, is required for example, for cost accounting and control. The entity type '*Work Centre*' usually represents this higher level. Of course, '*Work Centres*' are always defined per site, and '*Capacity Units*' are allocated to one '*Work Centre*' (an exception concerns the machine configurations discussed in paragraph 6311 and following).

6275 In an ideal data structure the other relationships with '*Site*' are optional. It can be useful to define a '*Bill of Material*' per site (for example material usage differs per site due to efficiency differences, or usage of different locally available components), and also to define a general '*Bill of Material*' that applies to all sites.

6276 In the RDM special attention should be paid to making '*Operation*' site specific, whereby complex routings can be defined across sites.

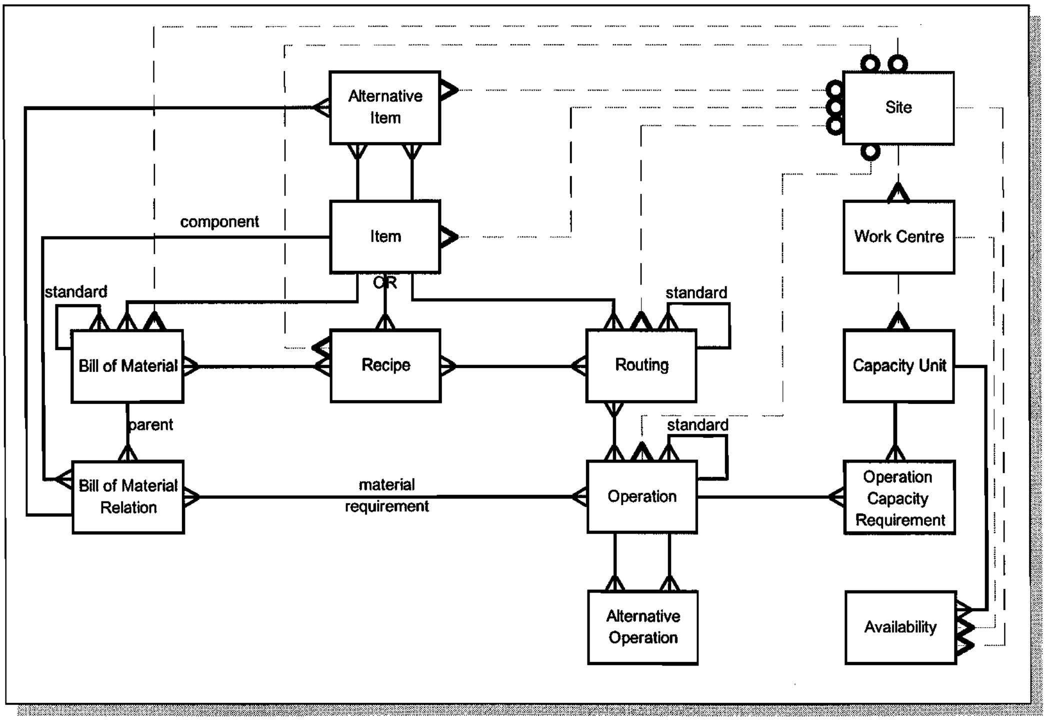


Figure 6.29 - Multi-site support

6277 Besides this multi-site structure, a number of BIS have a completely different realisation based on separate data structures per site. Data exchange then takes place via (intensive) message transfer. An important characteristic of such a structure is the physical redundancy of data. For example, the entity type *'Item'* is part of each separate data structure and individual items are recorded at every site (where relevant). During data maintenance, the BIS (or the database management system) has to monitor the consistency across the various sites. As this applies not only to items, but for many other entity types, this leads to intensive data transfer, which in turn makes high demands of the network. In return enormous flexibility is gained.

6278 Note that at a higher abstraction level one integral logical data structure certainly still exists. Behind the definition of the automatic data transfer there is a fundamental structure that controls the proper exchange of messages (this concept is based on the mechanisms of work-flow management systems). Essentially these implementations are by definition very dynamic and have the potential to change the data structure even in an operational environment. However, multi-site structures have only recently been introduced and still have to be proven in complex environments.

Units of measure and conversion

6279 The administration and reporting of *'Items'* may be desirable in various units of measure. For sales, purchasing, stock management, and subcontracting, different units can be required. Examples are: *number* (pieces, boxes, or pallets), *weight* (kg or tons), and *volume* (litres or m^3).

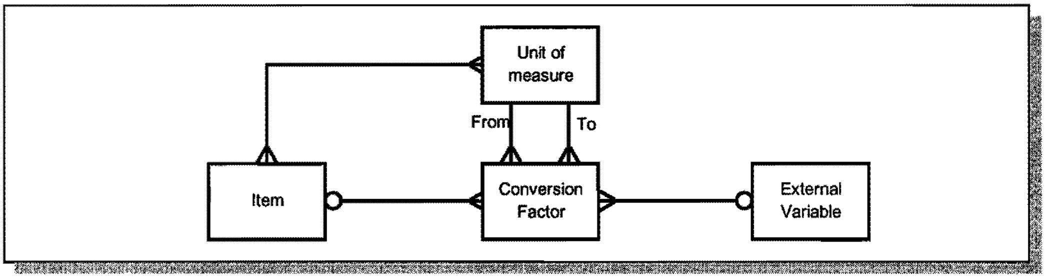


Figure 6.30 - Units of measure

6280 The conversion between units of measure can vary from being very simple, for example, from kg to ton; to complex, for example, from litre to kg. In the latter case the conversion factor depends on the specific *'Item'*. Finally, there are also conversion factors that depend on an external factor, such as the conversion from kg to litre for alcohol, which depends on the temperature.

Relation to purchasing and sales

6281 Here the relationship between 'State-independent Manufacturing Data' and the functional environment, in particular, the purchasing and sales functions will be discussed by examining the following aspects:-

(a) *purchasing and subcontracting*, where *'Item'* purchasing, hiring *'Production Capacity'*, and subcontracting *'Operations'* or complete production *'Routings'* are all dealt with;

(b) *sales and the relation with the customer*; here all of the customer-specific elements in the RDM are treated.

6282 Here also real BIS are projected onto the various parts of the RDM to illustrate existing BIS data structures.

Purchasing and subcontracting

6283 The relation with purchasing is present in every BIS, certainly regarding the purchasing of materials. In the typology of '*Items*' in paragraph 6226 and further, the type '*Purchasing Item*' is already introduced and defined. In figure 6.31 this entity type is shown next to '*Manufacturing Item*'. In the '*Bill of Material Relation*', the '*component*' relationship type refers to the general '*Item*'. The make-or-buy decision for an '*Item*' can then be delayed, depending on the in-house production capacity; therefore, an '*Item*' can belong to both types.

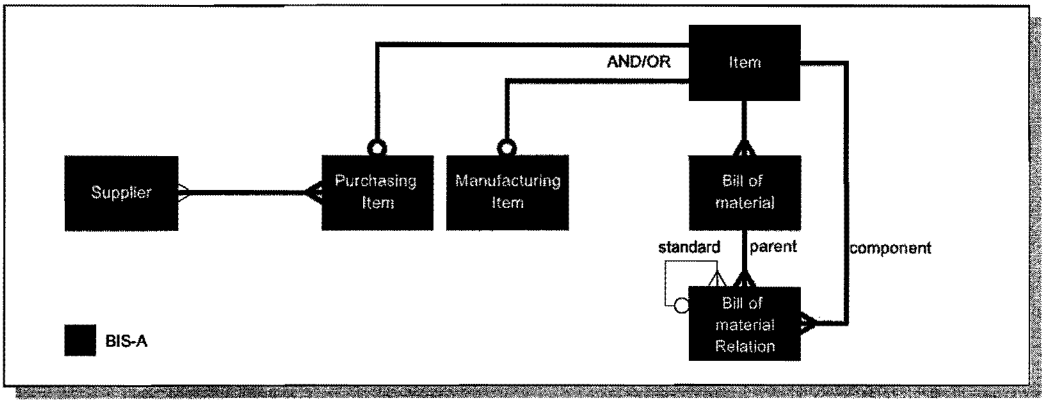


Figure 6.31 - Item purchasing

6284 The structure shown in figure 6.31 is actually implemented in virtually all BIS. The only difference may be in the implementation of the n:m relationship type between '*Purchase Item*' and '*Supplier*'. A number of BIS distinguish between one preferred supplier versus alternative suppliers. This distinction is discussed in the RDM for purchasing.

6285 Four forms of capacity can be purchased and are shown in figure 6.32, namely:-

(a) purchasing or hiring '*Production Capacity*'. The production is carried out in-house, but the tools, personnel, or other capacity is (temporarily) obtained from an external supplier. Typical examples are contracting specialised personnel and purchasing tools;

(b) subcontracting '*Operations*'. Here, part of the production is performed externally at the subcontractor's location. An essential difference with the normal purchase of the same semi-finished product is the supply of material to the subcontractor. In the RDM this material to be supplied is defined using a relationship type '*material requirement*'. There '*Capacity Requirements*' are not important, capacity is the subcontractor's responsibility (unless the subcontractor's capacity is planned by the subcontracting organisation, for example, if a certain total capacity is agreed upon)

(c) subcontracting all '*Operations*' for an '*External Work Centre*'. Here, decision to subcontract is taken at a higher level, namely the '*Work Centre*', which is directly related to this subcontractor (the supplier);

(d) subcontracting complete '*Routings*'. In this case the whole production is carried out externally at the subcontractor's location. The essential difference with normal finished product purchasing is again the supply of material and possibly the involvement or even responsibility for the capacity planning at the subcontractor. Generally this functionality is more important for trading companies than in a production environment, and is also called subcontracting from the warehouse.

6286 These capacity-oriented forms of purchasing are classified as '*Purchase Items*'. These are also included in the RDM for purchasing in chapter 5.

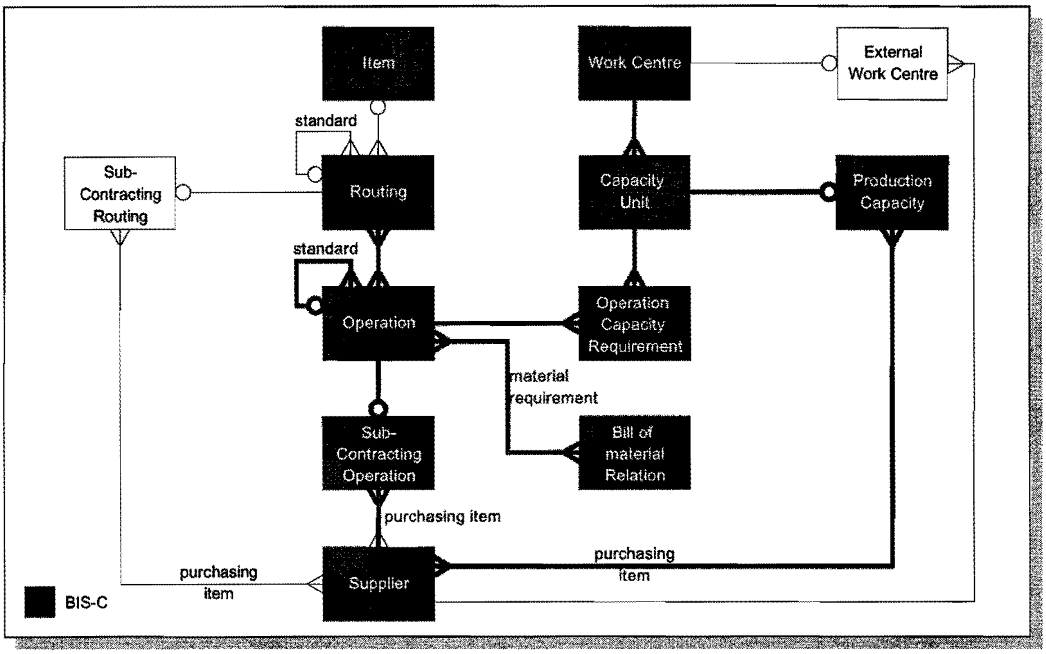


Figure 6.32 - Capacity and subcontracting purchasing

6287 Note that in the projection shown in figure 6.32, the entity type '*Production Capacity*' in this BIS should only be projected for tools. However, this is not clear in this projection because of the aggregated entity type (see also paragraph 6295, where the typology of '*Capacity Units*' is worked out).

6288 For subcontracting, a number of BIS do not go further than providing an 'external' indicator for the required capacity. In fact, subcontracting is then considered as virtually equal to the internal production. If this distinction is vague, then problems may arise with procedures related to material supply and the management and control of such (external!) '*Operations*'. In such data structures there is often no relationship with the supplier, which means the purchasing process is not automatically triggered.

Customer specialisation

6289 With customer order controlled production, entity types can be partially customer-specific ('*customer*' is an attribute type). In figure 6.33 the most important entity types, in relation to 'State-independent Manufacturing Data', are shown. A distinction must be made here between customer-specific and customer-order-specific data; although the latter form is outside the scope of this research, it can be stated that the data structure for both forms is very similar (only customer order dependencies, instead of customer dependencies).

6290 The customer-specific '*Items*' can be one of two types (see also the typology in paragraph 6226 and further), namely '*Variant Item*' and '*Special Item*'. The variants were discussed in paragraph 6257 and following, as the result of the configuration process on a '*Generic Item*'. These variants are always customer-specific.

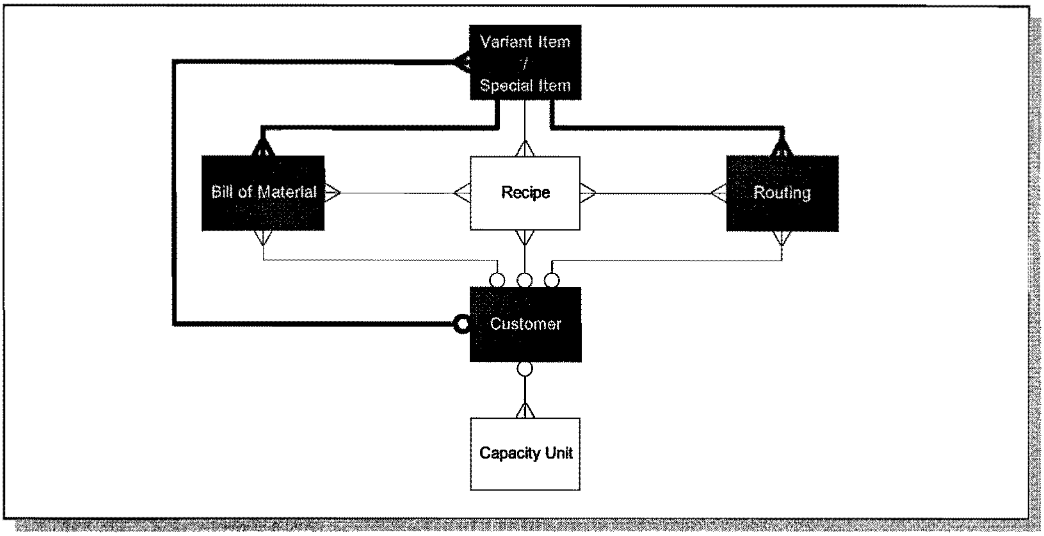


Figure 6.33 - Customer specialisation

6291 The other type, *'Special Item'*, does not go through the automated specification. These items should of course be specified, but then this is normally done manually. Usually such *'Special Items'* start as a dummy component in a customer-specific *'Bill of Material'* or finished product, and the specification occurs at a later stage. This results in extra attribute types such as the date the specification must be completed.

6292 Customer-specific *'Bills of Material'*, *'Recipes'*, or *'Routings'* can also be defined for *'Standard Items'*. These can be used to record elements differing from the normal structures, for example: customer-specific quality testing or components. In fact this kind of functionality can also be implemented using a *'Variant Item'*, the deviant elements turn the original *'Standard Item'* into a variant.

6293 Finally, customer-specific *'Capacity Units'* are also included in figure 6.33. These mainly involve tools such as moulds containing customer data that is imprinted on the finished product.

6294 In addition to these customer-specific data structures, order acceptance and control of the production process are the most complex aspects of customer order controlled production. The *state-independent* RDM provides little information on these aspects.

Typology of capacity resources

6295 In this section a typology of capacity-oriented resources is described, until now aggregated to the entity type *'Capacity Unit'* in the RDM. The following resource types will be discussed:-

(a) *single versus multiple resources*, where group structures of capacity resources are distinguished;

(b) *a distinction between resources in one or two dimensions*, which are defined and explained using examples;

(c) *an assignment of capacity resources based on skills* (or technology characteristics), where a more complex, dynamic form of capacity assignment is examined.

6296 This results in a matrix with the most important types of capacity-oriented resources. These are then expanded separately in the data structure of the RDM. Finally, an unusual organisation of '*Capacity Units*' is dealt with, namely '*Capacity Unit Configurations*' and in particular machine configurations.

Singular and multiple capacity resources

6297 The '*Operation Capacity Requirements*' are related to the entity type '*Capacity Unit*' in the RDM, see figure 6.34. The '*Capacity Unit*' is defined as the level on which Capacity Requirements Planning (CRP) is performed. Depending on the production control, these '*Capacity Units*' can be complex collections of the same sorts of machines, tools, or a shift of operators. However, at this planning level individual machines or specialised operators can also be defined as planning unit, for example, because they form a bottleneck in the production process.

6298 In the CRP, if production resources are planned and controlled as groups, or are not planned at all, then the final assignment of individual capacity resources must occur at the level Shop Floor Control (SFC). In figure 6.34 this distinction is worked out generically.

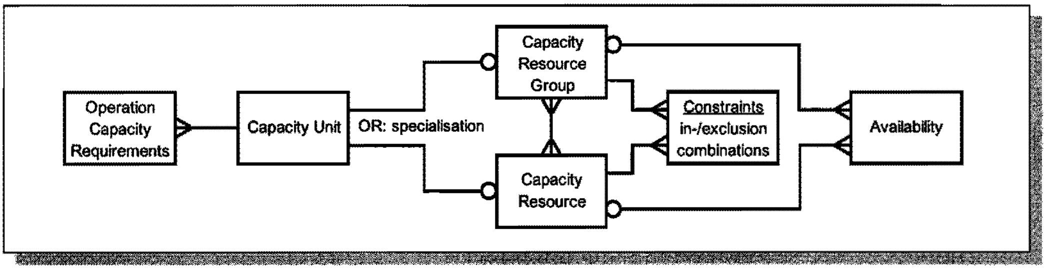


Figure 6.34 - Singular and multiple capacity resources

6299 In figure 6.34, both groups and individual capacity resources are specialisations of the entity type '*Capacity Unit*'. Examples are: shifts and operators, machine groups and machines, tool types and individual tools. The relationship type between them indicates that a '*Capacity Resource*' can belong to more than one group, for example an operator who (depending on the period) works in more than one shift. The availability of capacity resources is fixed in '*Calendars*', which can be defined both per individual resource (consider holidays or maintenance) and at group level.

6300 The '*Capacity Resource Groups*' can in practice be divided into two types, namely **homogenous groups**, in which all the individual elements are identical and completely interchangeable, and **heterogeneous groups**, where a preference exists (possibly situation dependent, for example the lot-size). For the homogenous capacity groups the final assignment can be easily carried out manually. The assignment for the heterogeneous groups can be simple but also very complex, and in some cases it can be worthwhile to use a decision support system. The decision rules concern, among other things, the combinations of capacity resources, such as certain tools only being applicable for certain machines, or operators are required with experience with certain tools. Therefore, by choosing a certain machine, there are direct consequences for the (remaining) choices of operators and tools. These '*Constraints*' are included here in the RDM. The decision rules can also be related to the skills of operators or the technology characteristics of equipment.

Skills and technology characteristics

6301 The decision rules for the assignment of *'Capacity Resources'* are sometimes based on desired skills or technical possibilities required by these resources. Depending on the nature of the *'Capacity Resources'*, these are called *'Skills'* (for operators) or *'Technology characteristics'* (for machines and tools). In addition to the skills required, *'Capacity Resources'* can contain these skills. This assignment problem can be solved relatively simply by using mathematical programming. These kind of algorithms are not present in most BIS or even in specialist shop floor control planning systems). The data structure associated with this is shown in figure 6.35.

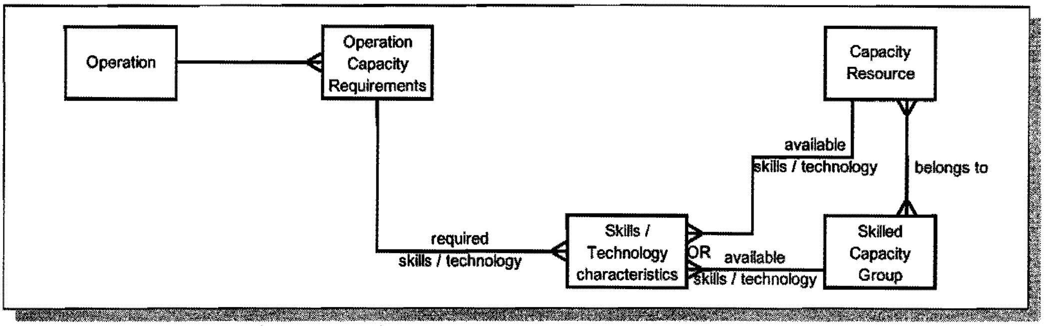


Figure 6.35 - Skills and technologies

6302 In practice the grouping of similar individual capacity resources into a *'Capacity Resource Group'* is usually worthwhile. Although this may lead to the inclusion of a resource in more than one group, this kind of grouping can make a vast simplification to the decision model when large numbers of resources are involved.

6303 There are only a few specialised systems that support the use of the *'Skills and Technology characteristics'*. The most well known example is FI-2 by IDS.

Capacity resources in one or two dimensions

6304 The most common capacity resources are one-dimensional. This means that a resource can be used in just one production order at a time. Therefore, this order possesses this resource completely for the duration of the operation. Thus the only dimension is 'time'.

6305 In addition, two-dimensional capacity resources exist. With these resources, more than one production order can be worked on at the same time, possibly under specific conditions. An example of a two-dimensional resource is an oven in which more than one product can be heated or baked at the same time. An important condition is the restriction on the second dimension, in this example, the number of m^2. Other conditions can be the same start time and/or stop time, or the same production recipe. In this example all products would have to be treated at the same temperature. Other typical examples of two-dimensional capacity resources are drying areas and storage areas. Power (such as electricity) is also sometimes considered a two-dimensional resource, with the second dimension being the maximum amount that may be consumed per unit of time.

6306 In data structures such a second dimension just means that an indicator with a maximum value has to be present as an attribute type in the *'Capacity Resource'*. In the *'Operation'* the amount required of this dimension should be defined. If the availability is not constant but changes in time, then this should be recorded as an additional attribute type in the entity type *'Availability'*.

6307 Besides the examples given, also '*Capacity Resource Groups*' can also be considered as two-dimensional resources. The capacity requirement for these resources is usually very simple, one unit during the operation time. The available capacity in the second dimension is defined as the number of individual resources.

Individual discussion on types of capacity resources

6308 On the basis of the previous classifications, table 6.2 gives a summary of the most important types of capacity resources that are recognisable in practice. For every type the format is shown in which they can occur, namely as 'singular' (as an individual resource), 'multiple' (as a group resource), skills or technologies applicable, and occurrence as a two-dimensional resource.

Capacity resource type	Singular	Multiple	Skills/Technology	Two-dimensional
Machines	YES	machine group	technology	only as a group
Operators	YES	shift	skills	only as a group
Tools	YES	tool type	technology	only as a group
Processing areas	YES	YES	characteristics	YES
Storage resources	YES	YES	characteristics	YES
Means of transportation	YES	YES	characteristics	YES
Power	YES	NO	NO	YES

Table 6.2 - Characteristics of different capacity resource types

6309 Some characteristics of these capacity resources are:-

(a) *machines* - with focus on:-

(i) although run time is an attribute type of '*Operation*', machines within one group can have different processing speeds. This means that it must be possible to adjust the run time based on an efficiency factor. In some semi-process industries the run time is not even relevant for the '*Operation*' because the processing speed is completely independent from the product and only depends on the machine. Further, the choice of tools can influence the processing speed, for example in die-casting. However, this relationship for '*Operation*' has already been modelled;

(ii) for maintenance activities it can be useful to work with a separate availability calendar per machine. If this is not possible, a good alternative is to register planned maintenance as a firm-planned order;

(b) *operators and shifts* - two (extreme) situations can occur here, namely:-

(i) *no special demands are placed on the skills of the personnel.* This means that *capacity is very flexible* and that fluctuations in capacity requirements can be absorbed with overtime or temporary personnel. This form therefore requires only minimum support, just a simple calculation of requirements;

(ii) *very special demands are placed on the skills of the personnel.* This means that after maximising the availability and skills of personnel (for example, through training), they are preferably split into skill groups. Such organisational measures can simplify the allocation of personnel skills that the complex data structure described above is no longer necessary;

because of holidays individual calendars per operator may be desirable for the more complex situations;

(c) *tools* - these are often not classified as a separate entity type, but are included as machines **or** items. However, there are a number of aspects that justify a separate classification:-

(i) tools can be purchased or have to be produced;
(ii) tools can wear out;
(iii) tools are (sometimes) structured as a '*Bill of Material*' of tool components;
(iv) tools can have a number of specific attribute types, such as delivery time, degree of wear, life span, and remaining life span;
(v) tools can be coupled to *machines*, *items* (one mould is used for a limited number of items), *operators* (because of skills with this particular tool, for example special forms of welding), and *customers* (because of customer information on printing plates);

figure 6.36 shows the data structure of '*Tools*' in the RDM.

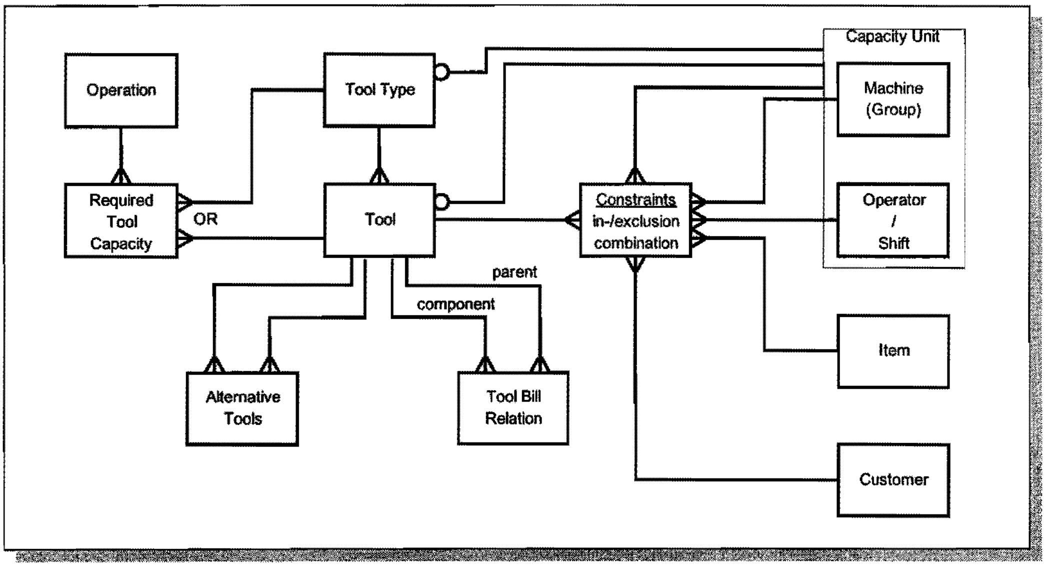


Figure 6.36 - Tools and their structure and constraints

(d) *processing areas* - some examples: ovens, drying areas and reactors. Some special features are:-
(i) the products undergo a treatment (as an '*Operation*'), in contrast to the following '*storage resources*', and these capacity resources are thus completely controlled within the framework of production orders (the 'filling' and 'emptying' both occur during the production of the order);
(ii) these capacity resources have a limited capacity, expressed in, for example, m^2, m^3, kg, or litre;
(iii) usually extra constraints hold, such as the simultaneous begin and/or end of all parallel production orders (and thus the same processing time), or the same processing recipe or setup (such as temperature);

such features are primarily recorded in the attribute types of these two-dimensional capacity resources;

(e) *storage resources* - some examples: silos, buffers, and containers. A few special features are:-
(i) these capacity resources are filled during the production process and have a limited capacity, expressed in, for example, m^2, m^3, kg, or litre;
(ii) with the exception of temporary buffers, these filled or partially filled storage resources may imply limitations for succeeding production orders;
(iii) these constraints lead to the need to model such storage resources as '*Capacity Resource*' for an '*Operation*';

(iv) the products, however, do not undergo any processing (*'Operation'*), which means that these products can remain in the storage location for a longer time, beyond the scope of the production order that 'filled' this resource;
(v) sometimes a limited variety of products is permissible, for example: within a given range or it is used for one particular product;
(vi) sometimes the storage resources should be cleaned or conditioned when products are changed;
(vii) when different batches are combined in one storage resource, the product features, such as potency and grades, should mix. This is, for example, relevant for the production of alcohol, where different batches of the raw material 'treacle' can have different qualities and sugar percentages, but must all be received in one tank;
(viii) sometimes there is a minimum and/or maximum storage time;

an important difference with the previous 'processing areas' is that control is usually beyond the scope and life span of the production order. This suggests using an extensive location system as part of the stock control function to manage these storage resources. However, when planning during the RCCP, CRP, and SFC, these constraints can also be critical. This leads to a relationship type between this two-dimensional *'Capacity Resource'* and *'Warehouse Location'*, see figure 6.37;

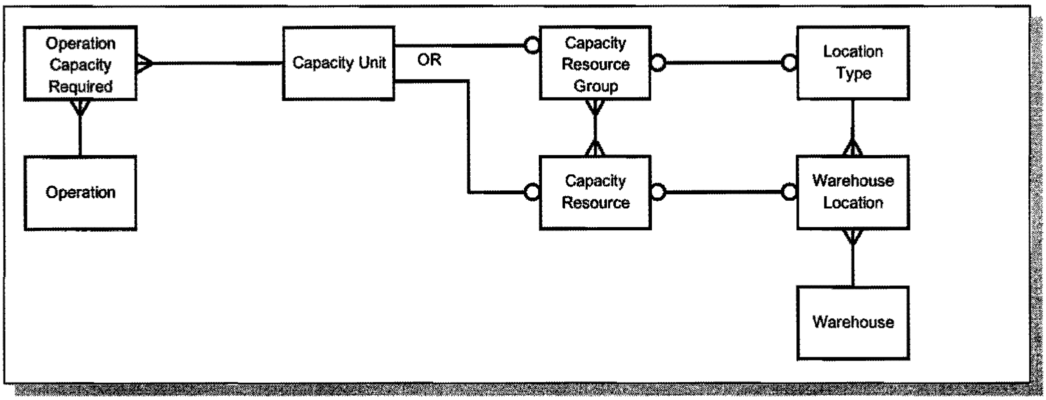


Figure 6.37 - Two-dimensional capacity resources as warehouse location

(f) *means of transportation* - with examples such as fork-lift trucks, AGVs, and conveyer belts. Further, the internal transportation can be considered as a normal *'Operation'* where the means of transportation behave as machines;
(g) *power* - with examples such as electricity or steam. Some special features are:-
(i) a limited utilisation per unit of time that behaves as a capacity restriction on this second dimension;
(ii) in contrast to the previously mentioned two-dimensional resources, the same start and/or end times are not required for the simultaneous production orders that consume this *'Capacity Resource'*.

6310 Summarising, the distinction between groups of capacity resources and individual resources is directly visible in data structures, just as the use of skills and technology characteristics is. This in contrast to the two-dimensional character of capacity resources, which is normally completely concealed in attribute types.

Configurations of capacity units

6311 In various production situations, the required machines are configured to form one physical unit because one or more '*Items*' are produced in the same way. Sometimes, all the machines and tools are moved to the same place, sometimes physical transportation links are created. While this configuration is being used, only products of this type or family of types can be produced. Because the turnover of these products is limited, this configuration is not permanent. One example is the production of snacks, where the proper machines and tools for mixing, shaping, and packaging are configured (see also [VLM93]). The data structure in the RDM is shown in figure 6.38.

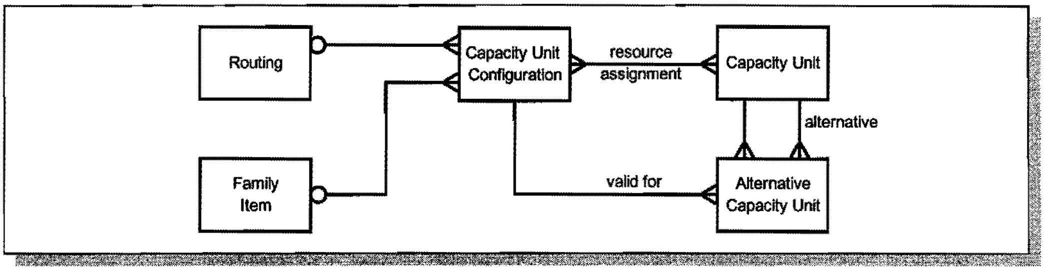


Figure 6.38 - Capacity unit configurations

6312 '*Capacity Unit Configurations*' can be linked directly to the '*Routing*' of the '*Item*' to be produced. If, however, this configuration is valid for a whole family of items, it is, of course, more efficient to relate it to the '*Family Item*'. A complicating factor is any '*Alternative Capacity Unit*' that can be included in a configuration.

6313 This support is very rare in BIS, with the exception of some specialised systems for shop floor control planning. This is also partly a result of the unavoidable but complex assignment problem of '*Capacity Units*' that can be required by more than one configuration at the same time.

Specialised / complex manufacturing

6314 In this section a number of subjects regarding special or specific features of some production processes are described. The following aspects will be discussed:-

(a) *diverging product structures*, mainly the consequence of co- and by-products as already defined in paragraph 6226 onwards;
(b) *multi-level 'Operations'*, the recognition of an extra level of '*Operations*', namely the '*Task*' as a lower level of control;
(c) *'Operation' and 'Task' planning*, where the diverging, converging, and network structures of '*Operations*' are discussed, and the possible forms of overlap concerning the sequential structures of '*Operations*' are dealt with;
(d) *support of sequence-dependent setup and change-over times of capacity resources*, especially depending on the items and their features;
(e) *distinction of different grades*, quality groups of '*Items*' that may overlap each other;
(f) *recognition of potency*, the percentage of an active ingredient for which the material requirement is determined.

6315 These special structures cannot be found in the average BIS, but must be sought in the more specialist, mainly semi-process systems. Here again, real BIS are projected onto the various parts of the RDM to illustrate existing data structures.

6316 The material structure in the RDM has until now been hierarchically defined. Namely the production of one finished product is performed on the basis of several components, and again each of which can also be built up from several components. There is, however, a large number of production processes which result simultaneously in more products than just that single finished product required. This multiple output concerns co- and by-products. These types of '*Items*' have already been defined in paragraph 6226:-

(a) *co-product items,* all primary products that are the required planned result of the production process. Depending on the planning mechanism, this primarily involves MPS-items;

(b) *by-product items,* all secondary products of any importance (because of the value or the environmental requirements) that are either an apparently unwanted result of, or remain after the production process.

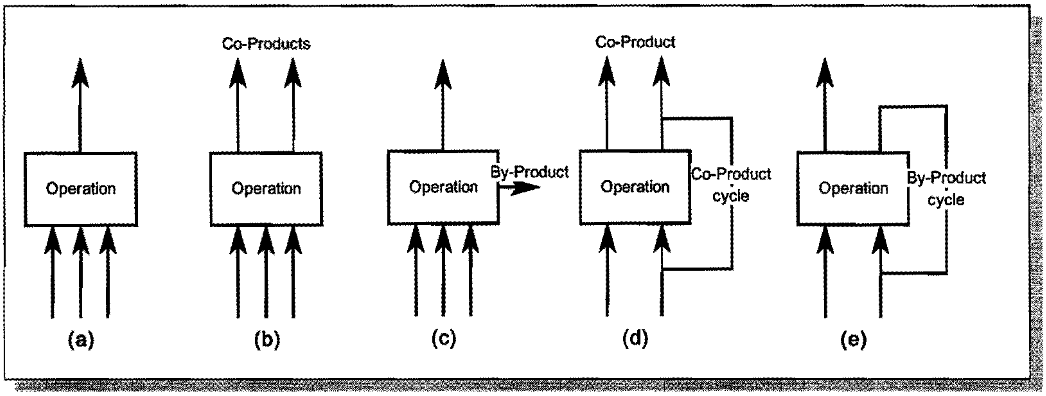


Figure 6.39 - Diverging product structures

6317 A short explanation of the structures of figure 6.39, see also [VLM93]:-

(a) the normal converging product structure is shown. In this example one parent item is produced using three component items;

(b) the production process results in two equivalent co-products. The amounts of these products produced is often a fixed ratio. Examples are separation or sorting processes, and some processes in the meat industry;

(c) in addition to the production of a primary product, one or more by-products are released (or remain after production). The relation between these amounts produced is normally known in advance. Examples are the livestock industry, where the primary products are meat or milk, however, manure is a by-product, which can be used as compost or for the production of biogas;

(d) in this structure several equivalent products are produced, one of which is partly fed back into the production process. An example is the production of yoghurt, where the primary product acts as a catalyst in the production process;

(e) the secondary by-product is fed back into the production process. A typical example is the use of catalysts.

6318 The classification above in figure 6.39 is based on one '*Operation*'. In practice, a '*Routing*' consists of a number of '*Operations*', and the production output certainly does not have to result from just the last operation (suppose the last operation is equipment cleaning). When several co- and by-products are involved, a final operation is often performed on just one of these products. This means that both co- and by-products can be the output of each phase or '*Operation*' in the production process.

6319 Figure 6.40 shows the RDM for co-products. The '*Set of Co-Products*' is central to this part of the RDM and it describes the group of primary products that are the output of a '*Recipe*' (or '*Bill of Material*' or '*Routing*', see paragraph 6202). The ratio in which these co-products are produced are also in this set. These ratios are not necessarily linear, which means that formulas must also be recorded in attribute types. The timing of the origination of the co-products is deduced from the relationship type '*co-product origination*'.

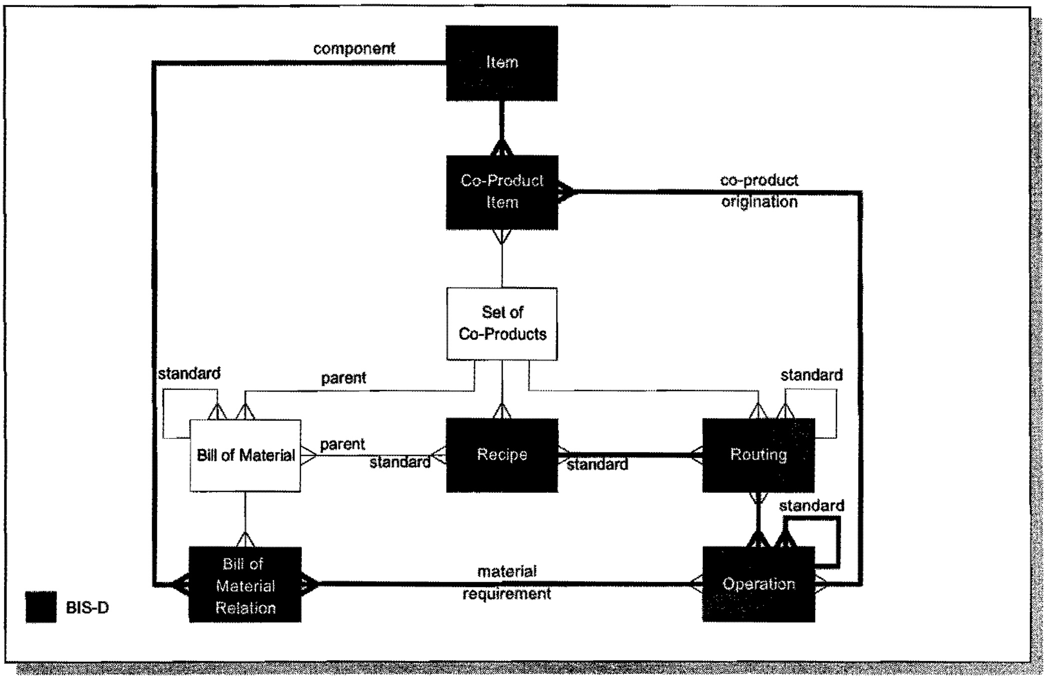


Figure 6.40 - Co-product structure

6320 A production process with by-products can basically be seen as a simplified version of the implementation of co-products. However, from the planning perspective, dealing with co-products is very complex. Certainly in the traditional MRP systems, it is virtually impossible to implement equivalent co-products without disrupting the entire material requirement calculation, which is the heart of the system. The calculation is much easier for by-products and is indeed present in various MRP systems. This less complicated structure is shown in figure 6.41.

6321 A separate entity type '*By-Product Item*' has been added to this part of the RDM. By-products are defined with a relationship to the material requirement structure ('*Bill of Material*'). The time at which these by-products are released is deduced from the relationship type '*by-product origination*'.

6322 In figure 6.41, comparable relationship types are indicated. By layering these relationship types, the solution for by-products in many BIS becomes clear, namely the common 'work-around' using a 'negative material requirement' or 'negative quantity per' for defining by-products. This similarity shows that both implementations are completely equivalent in terms of data structures. The extra complexity is in the underlying functionality of the BIS, which is not apparent from the data structure. This required functionality necessary in both implementations, is:-

(a) *the proper timing*, a normal material requirement is defined at the start of an operation. Without intervention a negative material requirement (the by-product) will also become available at the start of this operation instead of at the end of it. This creates a problem when processing by-products, as they have not been planned properly. A larger problem arises during the reuse of by-products within the same operation or production process: the cycles. If the by-product is both required and released at the start of the same operation in the MRP calculation this will result in a aggregated requirement of zero, which is of course incorrect;

(b) *the proper cost price calculation*, the by-products have a certain value, possibly negative if they have to be disposed of against costs. This value has an inverse influence on the cost price calculation.

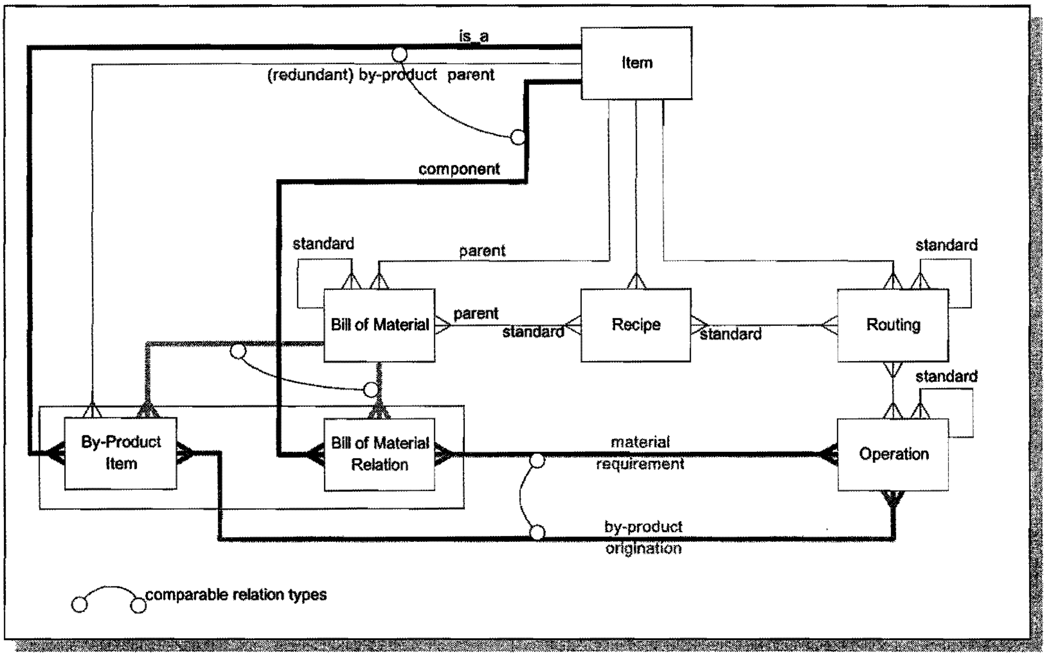


Figure 6.41 - By-product structure

6323 However, the problems mentioned above are solvable when certain assumptions are made, which is the reason why various MRP-based BIS are used in the semi-process industry.

Multi-level operations

6324 In some production situations an extra control level is required in the operation processing. Mainly in some BIS for the semi-process industry a third control level is available where operation characteristics can be defined, see figure 6.42.

6325 Material requirements are either defined at the middle level or at the bottom level, depending on the specific control situation. This is also true for the various forms of capacity requirements. Additionally, it can be worthwhile to apply a more complex operation planning at both levels. The possible planning forms are described in paragraph 6326 and onwards. Figure 6.43 shows the data structure of this part of the RDM.

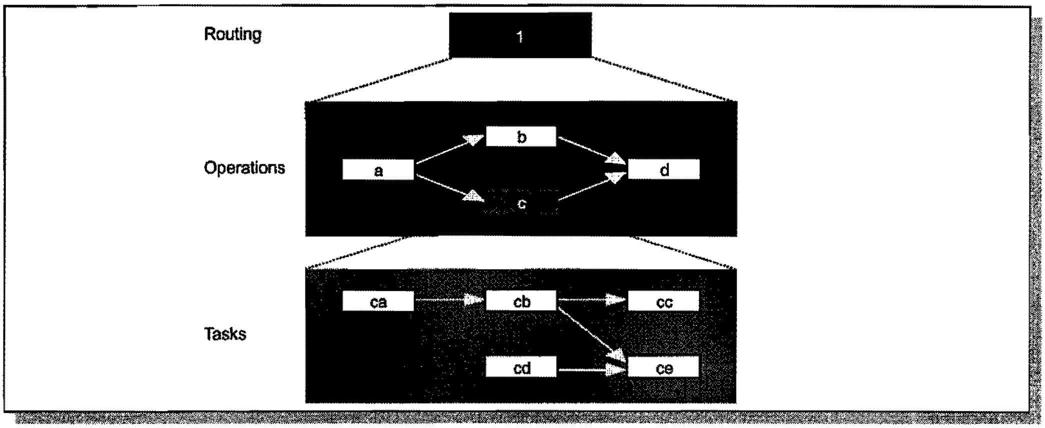


Figure 6.42 - Multi-level routing structure

Sequencing operations and tasks

6326 Many production processes are more or less sequentially organised, which means that all operations are performed one after the other. However, there are complex process structures where operations can or must be completely or partially executed in parallel. This can lead to a convergent structure of operations, where, for example, after a number of parallel assembly operations the semi-finished products can be joined together. Diverging operation structures also exist, for example, if the co- and by-products mentioned above each undergo a separate finishing process. For example, complete network structures for design and production operations can exist in complex 'engineer to order' situations. Figure 6.44 shows an example of an '*Operations*' network structure.

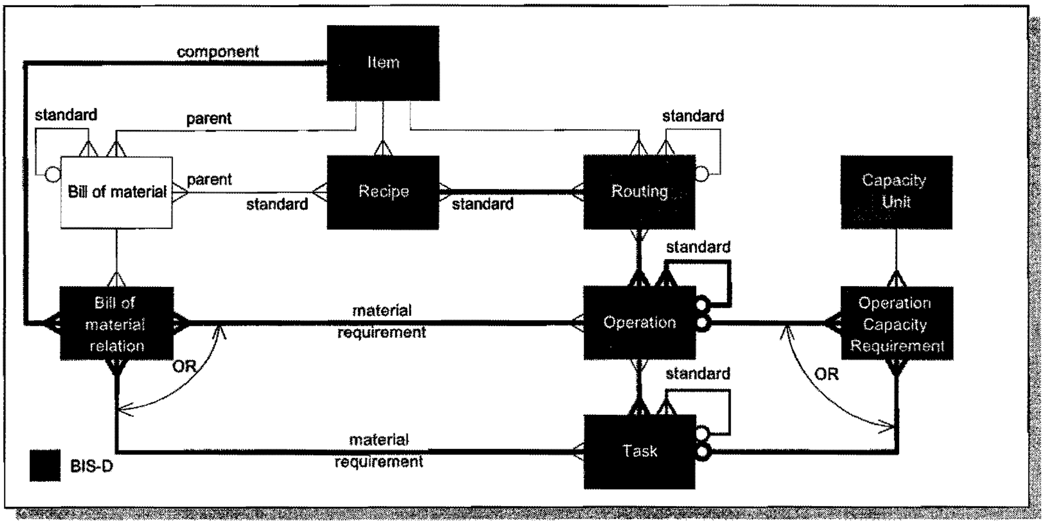


Figure 6.43 - Multi-level structure of routings, operations and tasks

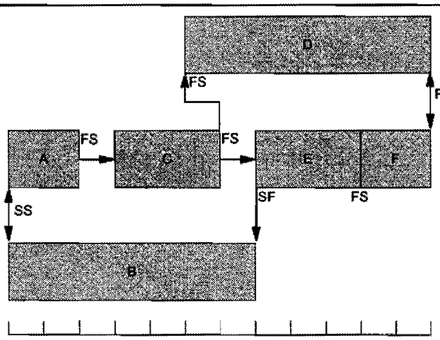
6327 Two techniques exist for establishing network structures: 'activity on arrow' and 'activity on node'. The second technique is consistent with the usual way of presenting networks of operations, such as in figure 6.44. The bars (nodes) of the network are the '*Operations*' (activities) and the arrows describe the relationships between the '*Operations*'. Four types of relationships between '*Operations*' are possible:-

(a) *finish-start*, this constraint entails a relationship between the start of the second operation and the end of the first operation. For normal sequential processes this relationship is implicitly present with a time value larger than or equal to zero. Particular examples are:-
   (i) *a minimum time*, for example, for the drying, cooling, transferring, or maturing of a product. This is also called 'cure time';
   (ii) *a maximum time*, for example, related to the dehydration or the shelf live of a product. This is also called 'pot life';
   (iii) *a constant time*, due to a combination of the factors given above;
   (iv) *a negative time*, where the 'next' operation may already start before the previous one is completed;

(b) *start-start*, this constraint entails a relationship between the initiation of two operations. In principle this defines parallel operations. Special examples are:-
   (i) *a time of zero*, so that two or more operations are carried out completely in parallel. This is also used as a 'work-around' to implement multi-resource scheduling;
   (ii) *a minimum time or amount produced*, for example, two operations are linked via a conveyor belt with a minimum buffer size;
   (iii) *a maximum time or amount produced*, for example, the conveyor belt has a maximum buffer size or, a transport batch is smaller than the production batch;

(c) *finish-finish*, this constraint entails a relationship between the completion of two operations. This can be the case in the situation in which two semi-finished products (A and B) are both needed at the same time for the next operation (C), but is usually solved by using two 'finish-start' relationships (A→C and B→C). This 'finish-finish' variation is more theoretical than practical;

(d) *start-finish*, this is actually identical to the 'finish-start' relationship.



Preceding Operation	Succeeding Operation	Relation Type	minimum value	maximum value
A	B	start-start	0	0
A	C	finish-start	0	999
C	D	finish-start	-1	-1
C	E	finish-start	1	1
E	B	start-finish	0	0
E	F	finish-start	0	999
D	F	finish-finish	0	0

Figure 6.44 - Sample network of operations

6328 The entity types '*Operation Sequence*' and '*Task Sequence*' use the attribute types '*relation type*' and '*relation value*' to describe the sequencing relationships. This second attribute type can describe:-

(a) a minimum value;
(b) a maximum value;
(c) a constant value, which can also be realised by setting minimum and maximum values equal to each other;
(d) a negative value;
(e) expressing production in time or in product quantity.

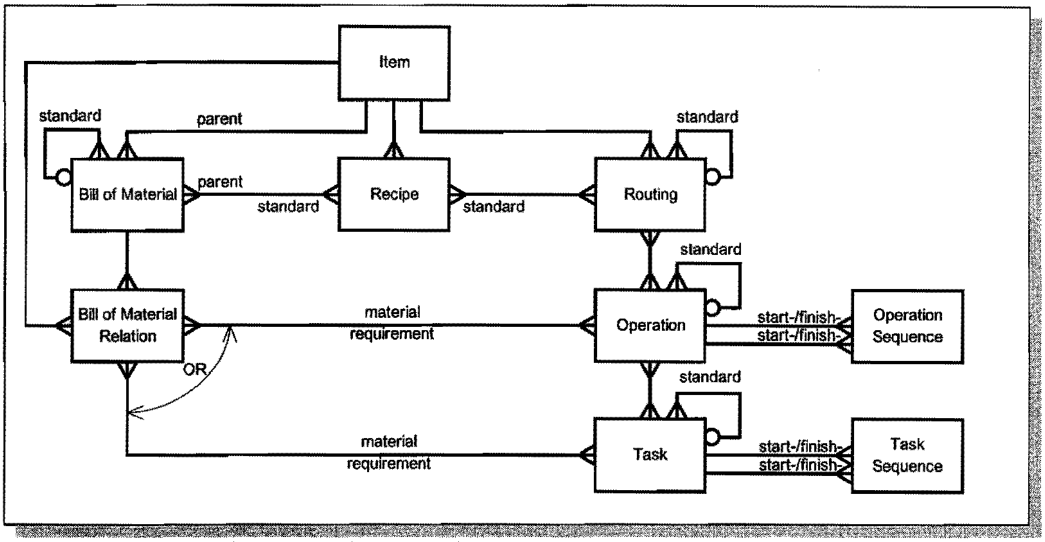


Figure 6.45 - Operation and task sequencing

6329 In many BIS, however, no such network structure is implemented, which is justified by the fact that most production processes have a much simpler structure. Two simplifications are used:-

(a) *a hierarchical structure of operations*, all divergent or all convergent. Because of this, no extra 'relationship entity type' is necessary and operations point to their successors in a divergent structure. The relationship type is usually a 'finish-start' relationship;
(b) *a sequential structure of operations*, however, expanded with an overlap parameter. This overlap can be defined in time, amount of product, or as a percentage. An overlap of 100% then means that the second operation is performed parallel to the first.

6330 In this last form, various attribute types are added to '*Operation*' so that the different relationships as described above can be realised. This results in several time variables, such as:-

(a) *a preparation time*, which can start before the semi-finished product of the previous operation has become available;
(b) *a change-over time*, during which the machine to be changed over must be available but the required specialist operators are not yet needed (the change-over is done by other operators). Neither does the semi-finished product from the previous operation always need to be present;
(c) *the actual run time*, during which all resources must be present;
(d) *the transport time to the next operation*, during which the production capacity is available for another operation;
(e) *the cleaning time of the capacity resources*, during which the production output is available for another operation.

6331 Some of these time factors can also be made negative as an extra 'work-around' to define other forms of overlap. The result is that such systems are capable of supporting very complex operation structures. However, this is a typical consequence of the continuous evolution of choices that were made during and after the original design of the system and which increases the risk that the system will disintegrate. On the basis of the previous analysis, one should either stay with a simple sequential operations structure or implement a real networks in the data structure.

Sequence-dependent setup and change-over times

6332 Setting up or changing over of machinery or tools can consume a certain amount of time at the start of an operation. This means that the setup time is an attribute type for the '*Capacity Unit*'. This setup time of a machine may differ, depending on the operation. In that case the attribute type moves to '*Operation*'. These are the simple and most frequently occurring forms of setup and change-over times.

6333 For example, for the production of dairy products (such as custard), the change-over time mostly depends on the previous operation. This usually results in standard production schemes, such as for from light to dark custard, or in other industries from wide to narrow, etcetera. Besides the planning based on such standard production sequences, more dynamic determination of production orders can be useful. On the basis of a matrix of change-over times from one operation to another it is possible, using mathematical models, to derive an 'optimal' operation sequence.

6334 The definition of these sequence-dependent change-over times can be based on the following aspects:-

(a) *items*, for example chocolate custard, yellow custard, and vanilla custard;
(b) *item features*, for example the colour in spraying;
(c) *combination of capacity resources,* for example the placement of certain tools in relation to a particular machine;

and naturally, a combination of these aspects is also possible. Figure 6.46 shows the corresponding data structure.

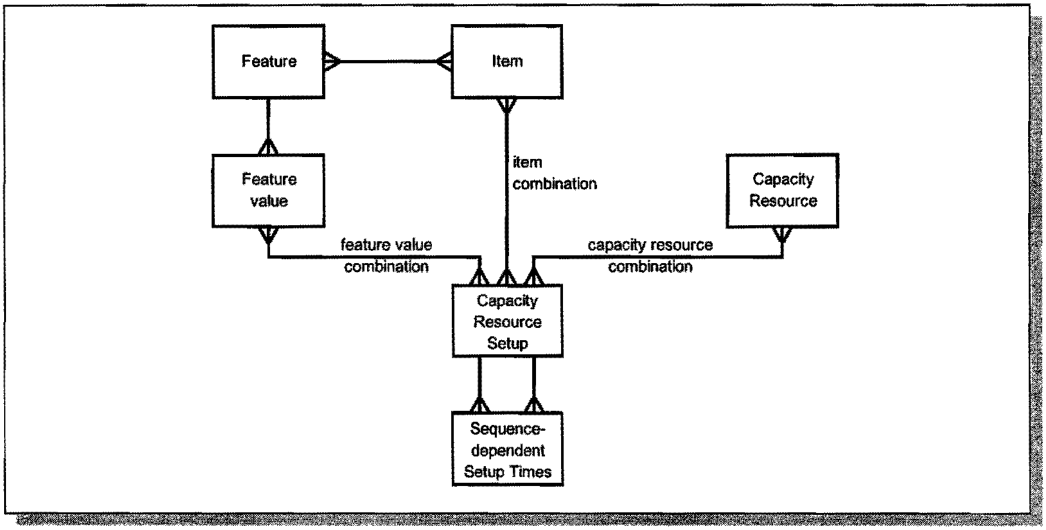


Figure 6.46 - Sequence-dependent setup times

6335 However, this functionality is not supported by the current generation of BIS, this is currently only available in the specialised shop floor planning systems.

Specification of item grades

6336 In the (semi-)process industry distinguish different qualities for a certain product is common practice. A simple example is described in [VLM93], where apples are processed into four finished products and the apples are classified into different quality classes ('*Grades*'):

	Apple quality top class	Apple quality medium class	Apple quality low class
fresh apples	OK		
apple compote in glass jars	OK	OK	
apple compote in tins	OK	OK	OK
apple juice	OK	OK	OK

Table 6.3 - Sample of grades

6337 The *'Grades'* can be defined in quantitative terms, for example, percentage X up to and including percentage Y, but also in qualitative terms, such as the apples in the example above. In assigning grades for foodstuffs smell, taste, and other testing panels are often used.

6338 The definition as separate *'Items'* seems to be an obvious solution for the implementation of *'Grades'*. However, this creates a number of problems:-
(a) when producing lots, the exact quality of the finished product is not-known in advance or the quality realised regularly differs from the quality planned, implying that these items must be repeatedly recoded;
(b) if these quality ranges overlap each other, a problem arises that can only be solved with extremely unnatural item definitions;
(c) often an enormous number of new *'Items'* are created that require maintenance and management together with their hundreds of attributes types.

6339 A better solution is the implementation using a separate entity type *'Grade'*. This part of the RDM is shown in figure 6.47.

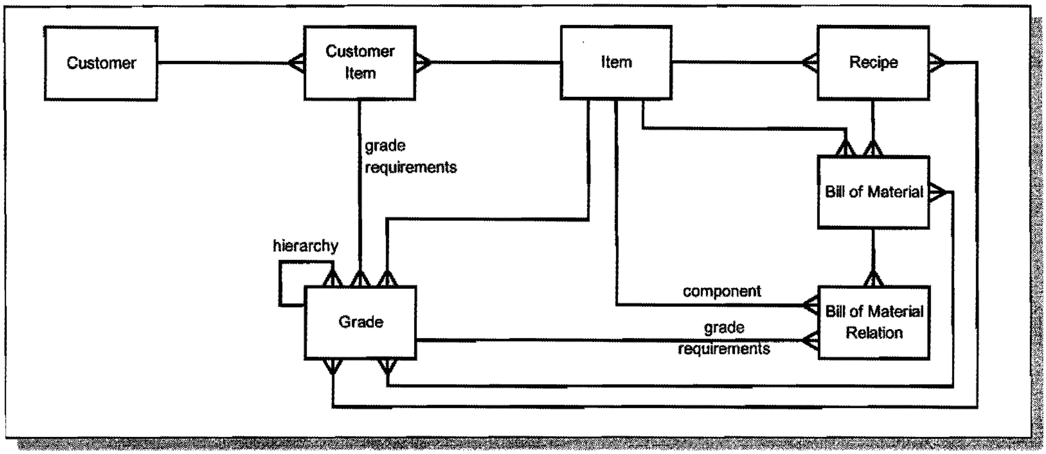


Figure 6.47 - Grades

6340 The lot quality should be controlled at lot-level. In the *'Bill of Material Relation'* the *'grade requirements'* should be defined, for the production of the given *'Item'*:-
(a) one *specific 'Grade'* is required;
(b) a *minimum 'Grade'* is required;
(c) a *range of 'Grades'* is acceptable;
(d) everything *outside a certain range of 'Grades'* is acceptable.

6341 These *'grade requirements'* also require a sequential or hierarchical structure of the *'Grades'* within themselves. Finally, the customer requirements in terms of *'Grades'* can also be relevant. A typical example of this is the production of alcohol, where certain qualities are even linked directly to particular customers.

Specification of potency

6342 A number of materials have a proportion of an active ingredient that basically determines how much can be used, this is called potency. Naturally, the heart of this problem is that this proportion or percentage can vary per batch. A fundamental difference with the grades mentioned previously is that the potencies can be converted.

6343 A good example is alcohol. The strength of the alcohol is indicated in percent. If a total amount of alcohol is required in the production process, it does not matter in principle which strength is used, as long as this is compensated in volume used. Of course this is only valid to a certain extent and certainly not for every process.

6344 In BIS this potency is only recognised in attribute types, primarily in the definition of the (state-dependent) batch. Usually for the *'Item'* and the *'Bill of Material Relation'* a normative percentage of 100% can be implicitly assumed, which is converted using the actual potency of each batch during the planning of production orders and the assignment of batches. In a number of BIS a different norm can be indicated for both state-independent entity types, for example, 'pressed out sugars 50%' for the raw material treacle.

Part 3

Test and Evaluation

7 Comparison between data structures and 'black box' approach

7001 The main objective of this research is to carry out an unbiased comparison of two essentially different approaches: the data-oriented approach and the 'black box' approach.

7002 From paragraph 7100 onwards this chapter discusses in detail how the comparison was performed. The steps described here further specify the research objective and approach given in chapter 3, as well as the motivation of the choices made.

7003 Then, beginning at paragraph 7200, a typical questionnaire (or checklist) is derived that forms the basis of the comparison.

7004 Final the results of the comparison are qualitatively and quantitatively presented starting at paragraph 7300. This is the basis of the final evaluation presented in chapter 8.

Specification of the comparison procedure

7100 The research approach was outlined from paragraph 3400 onwards. However, behind every procedural step in the comparison lies a number of choices aimed at guaranteeing an unbiased result. Each step is explained here and the choices, arguments, and consequences are discussed.

Definition of the functional scope

7101 The functionality provided by Business Information Systems is enormous, even in an area such as manufacturing (see also the definition of ERP in chapter 1). The figure below shows a modular view that covers most functional areas which can be required in a manufacturing company and may be available in a standard Business Information System.

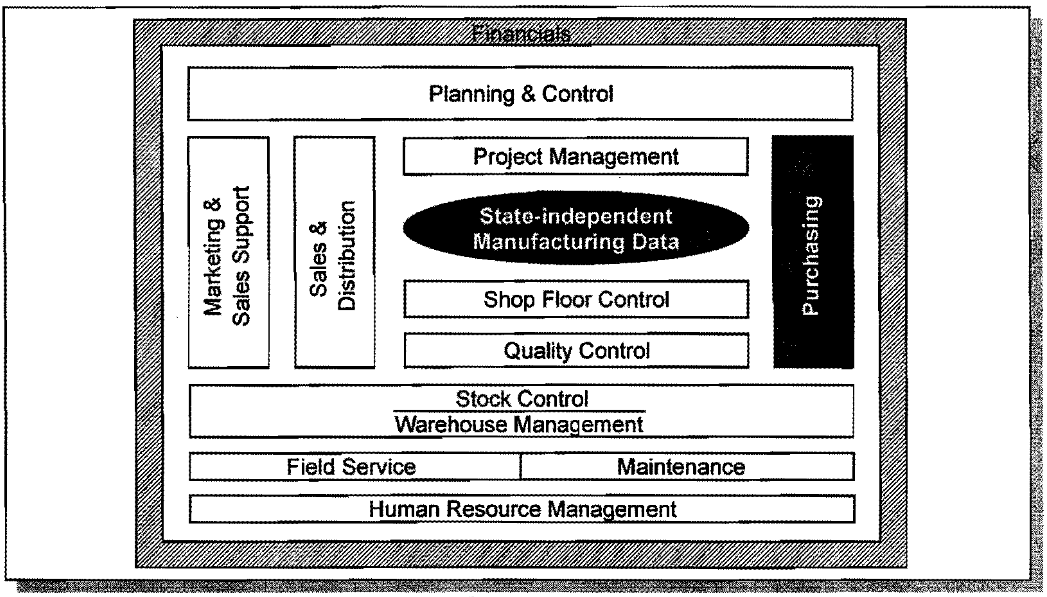


Figure 7.1 - Modular view on Business Information Systems

7102 There has been chosen for two representative areas from this broad functional spectrum:-

(a) *purchasing*, an area that seems strongly standardised in terms of information systems and is usually of an administrative nature. However, purchasing as a profession is emerging (and quite rightly so), resulting in a growing importance of tactical and strategic purchasing (see also chapter 5);

(b) *state-independent manufacturing data*, which describes all the specific characteristics of product structures and manufacturing processes. This area is highly company-specific and runs from 'make-to-stock' to 'engineer-to-order' and from 'discrete' to 'bulk' products (see also chapter 6).

7103 With these two strongly different functional areas most of the spectrum of business information systems is covered: the data structure of sales is more or less comparable with purchasing, and planning & control and shop floor control are both directly based on the (data) structure of the 'state-independent manufacturing data' (state-dependency only adds the dimension of time, which perhaps makes it more complex but not different in structure). In other words, the physical and functional characteristics of the primary processes are defined by means of the 'state-independent manufacturing data'. Furthermore, examination of the other functional areas is not expected to lead to any significantly different conclusions on the data-oriented approach.

Determination of the software packages to be examined

7104 To assess the value of the data-oriented approach and to justify the research objective it is theoretically only necessary to assess one of the many different software packages available.

7105 Hypothetically however, the conclusions may be influenced by the underlying technology of a system. In the beginning of the research a few older, third-generation systems (for example, developed in Basic without any DBMS) were examined, and conclusions drawn from the data structures had to be interpreted very carefully: the file structure hid many different record types and data elements that could be interpreted in different ways. Since the wide acceptance of relational databases, almost every modern software package has been based on this technology. Because relational databases require a more formalised design, the danger of misinterpretation because of 'tricks' (like in third-generation systems) has decreased. The consequences of future technological developments on the data-oriented approach will be discussed in chapter 9.

7106 In addition to the other support required from software suppliers, they must also be able to provide a data model of the software package. A few years ago this would have been a major problem for almost every available software package. Fortunately, nowadays more and more software suppliers have developed this important piece of documentation. The only hurdle is in obtaining such a data model, which is seen as the competitive advantage of the software package (which confirms the theory of the value of the data structure!). In all situations pertaining to the research purpose a non-disclosure agreement had to be signed.

Which software package do we choose to assess the purchasing function?

7107 Suppose a software package is chosen that covers all possible functions in purchasing, a situation which seems not completely impossible. In that case a comparison between the two approaches will lead to a limited result, since two situations may occur:-

(a) the data structure confirms the coverage of all this purchasing functionality. Naturally the software supplier will stress that 'everything is possible', which means the approaches will not come to different conclusions;
(b) the data structure is not able to confirm all this purchasing functionality because apparently there are some functions that do not affect the underlying data structure. Nevertheless, the data structure is not expected to deny this functionality and therefore no conclusion at all can be reached.

7108 This means that assessing a software package that covers all possible functionality in the area of purchasing will not add anything to the demonstration of the effectiveness of using data structures. Conclusions can only be related to:-
(a) the possible efficiency of using data structures, like compactness;
(b) the mere fact that such a software package actually exists, which is an interesting conclusion in itself.

7109 In light of the above considerations a software package[17] was selected, partly because of the fact that I could be certain of the co-operation of the supplier and partly because of the availability of the required documentation. The version of the software package examined was positioned by the supplier for 'make-to-stock' manufacturing (newer releases are also applicable for 'make-to-order' manufacturing). A focus on standardised (encoded) materials is therefore to be expected. The data structure of this software package was projected in the examples given in the description of the Reference Data Model for Purchasing (chapter 5).

Which software package(s) do we choose to assess the 'state-independent manufacturing data'?

7110 The 'problem' of software packages that may cover every possible product and process structure is not relevant here. Even for software packages that can handle many different manufacturing structures, choices will have been made as to the implementation of the product and process structures. In contrast to 'Purchasing', many different implementations are possible.

7111 Especially in the area of 'state-independent manufacturing data' it proves to be illustrative to consider different software packages. In view of the diversity of manufacturing processes supported, 'hidden' in the data structure, it is useful to compare these structures and to position the software packages with regard to each other. For example, the subject of co- and by-products can be implemented in very different ways, each one with its own advantages and disadvantages. Of course, the projection of different structures on the Reference Data Model (RDM) also demonstrates the use of the RDM strategy and the power of the RDM itself.

7112 In view of the availability of a data model, the supplier's support, and in particular the need for software packages that operate in different manufacturing situations, the following software packages were selected:-
(a) Triton 3.0, by Baan International. This Unix-based package focuses on discrete manufacturing (make-to-stock, assemble-to-order and make-to-order) from small manufacturing series to repetitive manufacturing;
(b) TXbase, by TXbase systems. This is also a Unix-based package and has particular functions for batch-mix and semi-process industries;

---

17 Because of the possible commercial sensitivity of the comparison results in this official publication the reducibility to the individual names of the software packages and suppliers has been omitted. The software package meant here is one of the four packages mentioned in paragraph 7112.

(c) Ratio, by JBA/Ratioplan Benelux. This AS/400 package is particularly well positioned in terms of combinations of discrete and semi-process industries;

(d) Prism, by MAR C.A.M. This is also an AS/400 package that is highly specialised in (semi-) process industries. It uses a very different structure and philosophy (like resources and production models) than most other software packages use.

7113 To illustrate these various characteristics, examples of projections of the data structures of these software packages have been included in the RDM description in chapter 6.

Composition of a typical checklist or questionnaire

7114 To compare the two approaches a representative and objective questionnaire (checklist of functional subjects) is required. Theoretically, one could set up a questionnaire that corresponds exactly with the data-oriented approach for example „Is a purchase order line directly related to a sales order line?". Questions like this however are unusual and practically awkward. In specific areas attempts have been made (see [LOG94.2]) but with unsatisfactory results.

7115 On the other hand, it is also possible to set up a questionnaire in which the data-oriented approach will lead to worse results or no conclusions at all for example only process questions like „When adding new materials are the accompanying ledger accounts also validated?". Both approaches can be expected to have their strong and weak points in terms of efficiency and effectiveness in drawing conclusions concerning the functionality offered by a software package.

7116 Bearing in mind the research objective, it is neither possible nor necessary to produce a 'complete' questionnaire. What is important, however, is to know the background and meaning of the various questions, particularly because quite often the primary difficulty with questionnaires concerns interpretation!

7117 It is not useful to include questions in the test for which it is immediately clear that they cannot be answered by data structure assessment. These will demonstrate the weaker points of the data-oriented approach but can be left out of the testing procedure with the different suppliers. A questionnaire that fulfils the requirements is derived in the next section. The questionnaire itself can be found in appendix C.

Drawing conclusions on the basis of projections on the Data Reference Model

7118 To compare the two fundamentally different approaches, conclusions must be drawn regarding the composed questionnaire, based on the data structure. The interpretation of the data structures of the software packages selected is based on their projections on the RDM, as illustrated in chapters 5 and 6.

7119 The development of an RDM is not trivial, and to a certain extent the process of projecting the data structures onto the RDM is difficult as well. On the other hand, this projecting process itself is not meant to be part of the research objective. Many conclusions can of course also be drawn from the data model of a system directly. The added value of the projections on the RDM is the addition of a context (that which is not covered) and the possibility to compare different software packages. The added value of the RDM will be discussed in chapter 8.

7120 In the case of questions directly concerning the singular data attribute types, the projection onto the RDM of course does not add anything. Those conclusions can be obtained directly from the data dictionary of the system.

7121 When interpreting the data structures of the software packages (or their projections onto the RDM) it is important to make statements concerning the potential functionality of a system. This functionality, however, does not have to be implemented (yet) in the software code. Although this seems an insurmountable problem, it is not, because:-

(a) the design of this data structure will not have taken place without a good reason;
(b) the definition and implementation of the right data structure itself is one of the most important steps in designing an information system, and this part has been performed;
(c) the presence of the correct data structure makes the development of the required functionality 'relatively' easy.

7122 For the purpose of this research project and also the application of the RDM in practice, the above-mentioned interpretation risk is limited. The objective is normally focused on finding the limitations of the data structure and software package, hence the functionality that the supplier claims to cover. Only the interpretation of the singular data elements should be handled with more caution.

7123 The conclusions drawn from the data structures are discussed in paragraph 7300 and further. Because the data structures themselves are an integral part of this discussion, they will be referred to in the questionnaire in appendix C.

Obtaining the supplier's reply on the questionnaire

7124 The resulting questionnaire is also formally answered by the software suppliers concerned. In fact, the same approach is followed as in one of the steps of a common software selection. This means that the suppliers respond with a rather brief reaction, usually confirming the presence or absence of the requested functionality.

7125 In this case, already published answers from these suppliers are used. This also means that an independent screening on the correctness of these answers has been taken place by the independent authors of this publication.

Confrontation of supplier and comparison of results

7126 The research thesis to be examined is that conclusions about software functionality based on data structures will be 'better'. By this I mean: more correct, more balanced and providing more understanding. To establish this, the statements based on data structures and the supplier's responses have to be compared and verified with the suppliers. Especially because the supplier is more or less biased, his confirmation of the resulting comparison gives a very objective view as to the correctness of the conclusions, for:-

(a) if the conclusions based on the data structures are better than the supplier's response, this means that with the supplier's approval he has to diverge from his previous reply on the questionnaire;
(b) if the results based on the data structures are not consistent with the supplier's response, the supplier will certainly not fail to mention this explicitly.

7127 The differences (and similarities) in results between the two approaches, as well as the supplier's feedback, have been categorised and are discussed later in this chapter. This categorisation particularly concerns a quantitative analysis of the differences.

Drawing conclusions as to all research objectives

7128 Chapter 8 is fully dedicated to the evaluation of all research objectives. Both strong and weak points of the data-oriented approach are discussed. The basis for this evaluation was:-

(a) the experiences with the development of the RDM;

(b) the lessons in interpreting data structures of software packages and projecting them onto the RDM;

(c) the conclusions from the direct comparison between the data-oriented approach and the 'black box' approach, as analysed in the rest of this chapter;

(d) the general practical experience in selecting and implementing standard business information systems.

Composition of a representative checklist or questionnaire

7200 To compare the data-oriented approach with the usual questionnaires to be filled out by a software package expert, a representative and objective questionnaire is required. Unlike a common software selection this comparison only concerns the functionality of the assessed software packages. All other (of course also very important) elements are outside the scope of this comparison.

7201 In this document as well as in daily practice, the terms 'checklist' and 'questionnaire' are used interchangeably. A (functional) requirement can be formulated as a question to be answered by the package expert. Checklists are normally used to draw up the requirements, which are translated into questions in a next step. Basically, checklists are very concise and the exact requirements will be established and formulated in a company-specific situation. The checklist is, therefore a tool that helps the expert in formulating requirements and questionnaires. For example, the checklist may contain an issue like 'subcontracting with material supply', which could result in questions like:-

(a) „Must subcontracting take place on the basis of a material code or an operation code?" [LOG94: 6.1.9];

(b) „Are two different delivery times available for the options: own manufacturing and subcontracting?" [LOG94: 6.1.10];

(c) „Is a procedure available for material delivery to subcontracting firms?" [LOG94: 6.1.11].

Available questionnaires (or checklists)

7202 Many attempts have been made to develop the ideal (company independent) questionnaire, but they have never succeeded because of the inevitable incompleteness. This quest only leads to unmanageable enumerations. That is why professional software selectors limit themselves to checklists with a concise formulation stating the most important issues to be reviewed. Therefore every consultancy firm has its own checklist that is treated cautiously (as being a competitive advantage) and is only meant for internal use. Hence only a few questionnaires are publicly available. Finally, I was able to collect the following questionnaires and checklists:-

(a) Guide to production control systems [COM79] and [COM81]. These are probably some of the oldest standard questionnaires on the basis of which more than 70 different software packages are compared. Although not adequate for our research purpose, they are interesting from a historical point of view;

(b) Lexitron [GAN83]. This questionnaire was developed as a result of several software selections. It is very much restricted to the requirements of the MRP-controlled companies involved. Furthermore, it is (of course) focused on the standard software and technology of a particular period (nineteen-eighties) and their limitations. Although this did not make the questionnaire very useful for our purposes, it is interesting to see which requirements were formulated at the time. For instance:-

  (i) the ability to store and maintain parent-component relationships: 'add, change, delete single-level BOM'; 'update where used when changes are made'; 'perform multiple deletes/additions, same-as-except, and multiple replacements'; 'fractional quantity per'; 'copy add capability';

  (ii) BOM reporting requirements: 'automatic printout of BOM when changes are made (on-line vs batch)'; 'single-level BOM'; 'single-level where used'; 'multilevel BOM or indented BOM'; 'multilevel where used'; 'summarised explosion/implosions'.

  Although still necessary requirements, no questions about them are currently included in any MRP package;

(c) Coopers & Lybrand International Manufacturing Systems Methodology [C&L80/85]. This is an extensive questionnaire that was used to evaluate various software packages, also in the eighties. It contains, for example, 70 questions concerning 'Purchasing' and 200 questions about 'State-independent Manufacturing Data'. Although it consists of many still relevant questions, it is strongly 'make-to-stock'-oriented and meant for discrete manufacturing situations;

(d) BuySmart!™ [BUY92/95]. This is an expert system developed in the USA (represented in the Netherlands by AVVA Advies), that also contains an extensive questionnaire recording functional requirements. This expert system contains 95 questions on the area of 'Purchasing' which is even more than in [C&L80/85];

(e) SAP-PPS: Kriterienkatalog [IPK94]. This is an unpublished questionnaire from IPK, a German consulting company, and is used for supporting their software selection projects. The questionnaire is very detailed for example, it contains 66 combined questions in the area of 'Purchasing' built up of 1 to 8 sub-questions;

(f) Coopers & Lybrand Netherlands Knock-Out Criteria [C&L94]. This internal checklist is only focused on possible knock-out criteria, which means it lacks the more detailed questions relevant to the final assessment of a software package. The criteria present are very up-to-date and significant in comparing the different approaches;

(g) Logiplan and Berenschot Research on Packages for Manufacturing [LOG94]. This is an investigation on the properties of software packages available on the Dutch market (although most of them concern the main international players). The investigation has been performed every two years over the last ten years. The questionnaire was revised thoroughly each time and is probably one of the best thought out and up-to-date questionnaires available.

7203 A short analysis of all these questionnaires shows that they (of course) overlap each other a lot. Most of the issues addressed are all very relevant to specific manufacturing situations. On the other hand, none of the questionnaires is even close to completeness.

Design criteria for the questionnaire

7204 Researching software functionality is not just a matter of using existing questionnaires or just adding as many questions as come up (nor is it the right way for selecting software). The research objective is not to focus on a rough assessment, like the availability of modules or the coverage of knock-out criteria, but it goes at least one level deeper and chooses to assess the preferred or selected package in detail in order to really understand the possibilities and limitations of the software.

7205 Software packages have changed in time. More and more functionality is covered and technology is also becoming more and more flexible. This has resulted in a shift from 'can a certain report be produced' to 'is certain data available'. Nowadays questions will therefore be formulated quite differently from those in the past. Design criteria have been divided into two questions:

About which functional areas is detailed knowledge needed?

7206 The RDM in chapters 5 and 6 covers two broad functional areas that are described in terms of data structures. First, the questionnaire composed must cover these areas and be well balanced as to all functionality. Based on the structure of the questionnaires examined the following subdivision can be made:-

(a) purchasing:-
  (i) state-independent (including suppliers data, pricing structures, alternatives);
  (ii) state-dependent (covering the full process of purchasing: requisitions, quotations, contracts and orders, receiving, invoicing, evaluation);

(b) state-independent manufacturing data:-
  (i) material definitions (including classification, alternatives, measure conversions);
  (ii) product structures (including bills of material, co- and by-products, product configuration);
  (iii) resource structures (including work centres, tooling);
  (iv) process definitions (including routings, production models);
  (v) engineering changes (including change policies, drawings, mass changes).

What type of questions can be formulated?

7207 The questions should not be just data-oriented or just process-oriented (see the description of the difference between 'black box' questions and 'internal structure' questions starting at paragraph 3200). Based on this analysis distinction is made between:-

(a) data-addressing questions, which can be formulated directly (asking for certain data attribute types, entity types or relationship types) or indirectly, but still addressing data elements;

(b) process-addressing questions.

This distinction has been further detailed below into a classification that is useful for the research objective.

Data-addressing questions

7208 Questions directly or indirectly addressing data stored by the software normally only concern the static characterisation of a system. Typical questions are: „is certain information stored or available". All data-related questions can of course be expected to have a direct relation with the data structure. To classify these questions the following subdivision has been made:-

(a) *singular data*, concerning data attribute types. For example:-
  (i) „Is it possible per purchase order line to state a separate delivery time?" [LOG94: 6.1.3];
  (ii) „Is it possible to have multiple descriptions per item?" [LOG94: 3.1.2];

(b) *multiple data*, concerning data entity types. For example:-
  (i) „Are tools recorded in: item master file, machine file, separate tooling file?" [LOG94: 4.2.1];
  (ii) „Are blanket orders and call-offs possible?" [LOG94: 6.1.5];

(c) *complex data*, concerning data relationships between entity types. For example:-

(i) „Can purchasing take place on (customer) order (item not recorded in the item master file)?" [LOG94: 6.1.2];

(ii) „Are tools with a relational record (M:N) directly linked with an operation, a machine, a product, a customer?" [LOG94: 4.2.3].

7209 As mentioned before, a distinction can also be made between direct and indirect formulation. Some examples:-

(a) *directly*:-

(i) „Is it possible per purchase order line to record the: required, promised, realised delivery time?" [LOG94: 6.1.4];

(ii) „Is it possible to record different potencies or grades for a given item per item code, multiple item codes, item + lot number, item + release number?" [LOG94: 3.1.5];

(b) *indirectly*:-

(i) „Must subcontracting take place on the basis of: material code or operation code?" [LOG94: 6.1.9];

(ii) „Are two different delivery times available for the options own manufacturing and subcontracting?" [LOG94: 6.1.10];

(iii) „Is monitoring the supplier quality (vendor rating) possible for: delivery time, quality, delivery quantity, or none?" [LOG94: 6.1.14];

(iv) „can a report be produced containing .....?";

(v) „can a history of .... be stored?".

7210 The difference between direct and indirect questions is just a matter of expression and not essential to the comparison. On the other hand, very clearly (direct) formulated questions may avoid any misunderstanding. For our purposes this difference has only limited relevance so long as the meaning of each questions is clear.

Process-addressing questions

7211 Questions concerning the processes supported by the software deal with the dynamic features of a system. Typical questions are: „is a certain action possible or procedure available". These questions can only be expected to ('hopefully') have some relation with the data structure. To classify these questions the following subdivision has been made:-

(a) *process presence*, for example:-

(i) „Is an integrated purchasing procedure available?" [LOG94: 6.1.1];

(ii) „Is a procedure for subcontracting available?" [LOG94: 6.1.8];

(b) *process step presence*, for example:-

(i) „Is a procedure available for material delivery to subcontracting firms?" [LOG94: 6.1.11];

(ii) „Is a procedure in place for inspection of incoming goods?" [LOG94: 6.1.16];

(c) *process step sequence*, for example:-

(i) „Can these delivery schedules be changed afterwards?" [LOG94: 6.1.6];

(d) *process step behaviour*, for example:-

(i) „Is there a standard provision for electronic receipt and transmission of purchase orders, call-offs, order confirmations, invoices?" [LOG94: 6.1.19];

(ii) „When adding new materials are the accompanying ledger accounts also validated?" [LOG94: 3.1.1];

(iii) „Can changes (mass replace) be made to bills of material and bills of operations in terms of: 'when the old material is finished', 'as soon as the new material is available' ... ?" [LOG94: 4.4.2].

Resulting questionnaire

7212 The classification under consideration is presented in table 7.1. The questionnaire cited ([LOG94]) is included and supplemented with the new questions in the area of 'Purchasing' from the 1996 edition ([C&L96]). An overview of all questions is presented in appendix C together with the class assignment of each question and a reference to the discussion of the subject of inquiry in the RDM, thus as it relates to data structures. If no reference has been included, then the RDM does not assess this subject.

	Data addressing questions			Process addressing questions			
	singular (data elements) **DS**	multiple (data objects) **DM**	complex (data relations) **DC**	process presence **PP**	process step presence **PSP**	process step sequence **PSS**	process step behaviour **PSB**
State-independent Manufacturing Data							
Material definitions	5	4			1		2
Product structures	7	2	6	1			2
Resource structures	8	5	7				
Process definitions	10		7				2
Engineering changes			3		1		3
Purchasing							
State-independent	5	1	1				
State-dependent	10	5	6	3	3		1

Table 7.1 - Categorisation of the questionnaire according to the 'data-process' typology

7213 A few remarks about the categorisation must be made before going into detail about the questions selected from the resulting questionnaire.

7214 The distinction between process- and data-oriented questions appears in general to be quite simple. The 'data-addressing questions' are recognisable from verbs such as 'to define', 'to record', and 'to relate'. In contrast, the 'process-addressing questions' are usually formulated as: „is something happening when ...." or „is a procedure available for...". A limited number of questions are more difficult to categorise because they are formulated like a 'process-addressing question', but with a one-on-one relation to data elements. If a direct reference is indeed intended then these questions are classified as 'data-addressing question'.

7215 During the subclassification of 'data- and process-addressing questions', the problem of multiple questions can occur. These questions can concern, for example, attribute types, as well as entity types, even sometimes also the relationships between entity types (for example LOG94: 4.2.5]). Those multiple questions are broken down into separate questions, which are then treated separately and classified in the appropriate class.

7216 A few things stand out when the resulting class assignments in the matrix are considered. First of all, as a consequence of the division of the questions over these categories, only a few questions remain per category. The scopes of 'Purchasing' and 'State-independent Manufacturing Data' are nevertheless very broad; see chapters 5 and 6. It is certainly possible to formulate a larger number of meaningful questions for all the classes. However, the division as presented is adequate for the current comparison.

7217 On the other hand it is also obvious that although this is an (extensive) standard questionnaire, large parts of the RDM are not dealt with at all; refer to subjects out of the RDM presented in Chapter 6, such as the typology of 'Bills of Material' and 'Routings', the typology of 'Items', the multi-site structures, the engineering documents, etc.. Questionnaires covering the complete area of 'State-independent Manufacturing Data' or 'Purchasing' are apparently much larger.

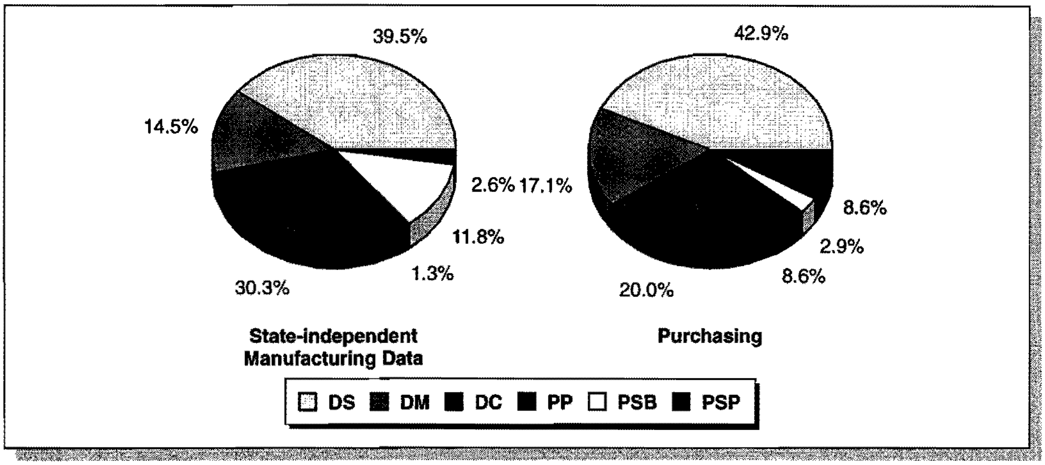


Figure 7.2 - Distribution of questions over the 'data-process' typology

7218 From figure 7.2 it can be seen that the number of 'process-addressing questions' is relatively small. This might partly due to the fact that the people developing the questionnaire designed it from the perspective of an expert.

7219 In particular, the number of 'process-addressing questions' is limited in the state-independent areas. Although this seems logical and was to be expected, it is actually possible, as well as worthwhile, to pose such questions. Examples are primarily related to:-

(a) simple and user-friendly maintenance of this (state-independent) data;
(b) control and clean-up of files;
(c) automatic generation of derived (state-independent) entities (such as the generation or configuration of routings);
(d) verification, authorisation, and monitoring of the consistency of this data;
(e) mathematical formulas, such as conversion of units-of-measurement or currency conversion.

7220 The justification for the low number of such 'process-oriented questions' lies in the limited degree of mutation with respect to the dynamic (state-dependent) data. The reasons for making these process-related demands are: a particular interest in authorisation or consistency, the size of certain data files (such as many tens of thousands of addresses, articles, or bills of material), and the necessity of large-scale mutation of core data on a regular basis.

7221 In principle, all of the questions from [LOG94] have been included in this research. An important reason for using this questionnaire is the fact that all of the answers to this questionnaire provided by suppliers have been published and are therefore verifiable.

7222 Aside from the questionnaires that were used, it would also have been possible to use the other questionnaires that have been mentioned here. However, examination of these questionnaires shows that their questions are completely comparable to those used, and that they therefore will not lead to other conclusions.

Comparison of supplier reaction with data structures

7300 The questionnaire employed, which is given in appendix C, has already been completed by suppliers as part of an official publication [LOG94]. In this research the testing and answering was carried out using data structures, which resulted in a significant number of differences that were explained in a confrontation with the supplier.

7301 The testing can be summarised as follows. First of all, the criteria on which the answers of the suppliers were determined to be correct or incorrect is examined, in the course of which especially the interpretation problem emerges. The results are then summarised quantitatively. In addition, several conspicuous matters with respect to the answers of the suppliers are described that were observed during the comparison of the data structures. These points are treated in detail since it is precisely these points that characterise the difference in approach. Finally, the results are analysed in relation to the 'data-process' typology of questions.

Criteria for the verification of answers: what is correct and incorrect

7302 The formulation of unambiguous questions is not easy and is frequently underestimated even by experts. Suppliers will have attempted to answer the questions in the way most beneficial to their situation. During verification based on data structures the actual goal of the question is focused upon and other interpretations are judged to be incorrect. Therefore, whereas poorly formulated questions in a standard publication will be explained to the advantage of the supplier, this research is concerned with the exact functionality and will not give the supplier the benefit of the doubt on what is really meant.

7303 A clear example is [LOG94: 4.3.13], „Are alternatives with different operation or lead times possible for different machines (same operation)?". This concerns alternative machines that could be used in an operation. This functionality was confirmed by one of the suppliers because a machine group made up of interchangeable machines can be recorded for an operation. This is, however, not the intention of the question; in this manner such alternatives cannot lead to deviant 'operation times' or 'lead times' and are therefore not operation-specific. Such conclusion follows directly from the data structure and leads to the identification of the original answer as being 'incorrect'.

7304 In this research there is a large difference between the demonstration of 'presence of functionality' and 'absence of functionality'. The questions [LOG94: 3.2.12] to [LOG94: 3.2.15], concerning variant bills of material and a product configurator, are a simple example. It quickly becomes obvious if such functions are missing, because of the absence of the various entity types (and relationship types). If some sort of generic material structure is present, then all elements of the data structure must be examined in order to answer each of the questions.

7305 This also holds for (direct or indirect) questions about attribute types. Failure to recognise this kind of attribute type leads to the conclusion that the underlying functionality is not supported. If, however, an attribute type is encountered that 'resembles it', it does not necessarily mean that this is the correct interpretation of the attribute type involved and that this functionality has also actually been implemented.

7306 For this reason a distinction is made in the quantitative analysis between testing regarding the presence of functionality and that regarding the absence of it; in other words, the confirmation of the 'Yes's' in relation to the 'No's' in the questionnaire. Furthermore, the answers that cannot be reliably verified are classified as '?'.

Quantitative results of the testing

7307 The results of the comparisons of the various BIS are presented together in table 7.2. In this table the questions concerning the 'State-independent Manufacturing Data' for the four BIS are given together with the questions regarding 'Purchasing' of a single BIS.

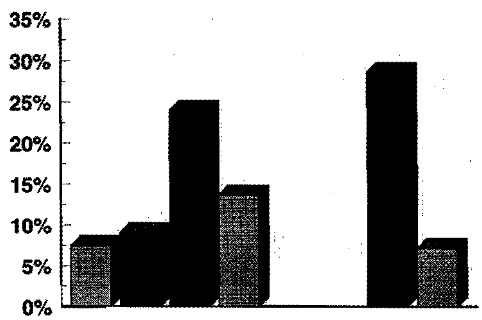
Type of questions	Number of questions	Presence of functionality	Absence of functionality	Number of errors found
Data Singular	135	55 + 16?	53 + 2?	9
Data Multiple	50	31	15	4
Data Complex	99	44 + 3?	28	24
	284	226 (80%) + 21? (7%)		37 (13%)
Process Presence	7	4	3	
Process Step Behaviour	37	2 + 23?	8 + 4?	
Process Step Presence	11	3 + 8?		
Process Step Sequence	0			
	339	246 (72%) + 56? (17%)		37 (11%)

Table 7.2 - Verification of functionality and number of observed errors

7308 Although it is perhaps already obvious, it will be emphasised here that the observed 'errors' do not result from deceptions by the suppliers, but rather have various other causes such as alternative interpretations and different versions. In fact, it was striking how accurately most suppliers actually fill out this kind of questionnaire, in spite of their commercial interests.

7309 Partly due to this accuracy, the observed 'errors' are usually the best indication of the benefit of using data structures. This claim is further strengthened by the fact that the answers in this questionnaire are screened beforehand by independent experts. These experts were therefore apparently unable - due to limited time and/or resources - to point out these (remaining) errors.

7310 The number of observed errors is given in figure 7.3 as a percentage of the total number of questions (per type).



Type of question	Manufacturing	Purchasing
Data Singular (DS)	7.5%	0.0%
Data Multiple (DM)	9.1%	0.0%
Data Complex (DC)	23.9%	28.6%
Total data adressing	13.7%	7.1%

Figure 7.3 - The number of observed errors compared to the number of questions per type

7311 No errors were found with regard to 'process-addressing questions'. This is partly due to the relatively low number of questions; however, it also serves as an indication of a weak side in the data-oriented approach. As can be seen from the number of '?' in table 7.2, the data-oriented approach is in many of these cases incapable of confirming the presence or absence of functionality, not to mention detecting errors.

7312 The observed errors are apparently chiefly associated with the questions of type DC ('complex data-addressing questions'). This indicates that the chance of errors is larger for more complicated functionality, but also that the approach based on data structures is apparently capable of detecting such errors effectively.

Notable differences and observations resulting from the comparison

7313 Before analysing the results based on the 'data-process' typology, the notable differences brought out by the comparison will be summarised. These differences give a good indication of the possibilities (and limitations!) of the data-oriented approach.

7314 Based on analysis of the data structure, several kinds of functionality indeed appears to be present, as indicated by the responses of the suppliers, but only in a certain context. These include among others the use of 'purchasing delivery schedules' and the use of 'a *requested* delivery date' in addition to 'the agreed upon delivery date'. Both turned out only to be possible in a certain BIS as extensions of the MRP functionality. Therefore, for example, 'purchasing delivery schedules' are not possible for requirement generation that is not MRP-controlled (but for example SIC-controlled)!

7315 In two cases, certain functionality was suggested based on the data structure that was shown afterwards *not* to be supported by the software; namely, the use of alternative work centres and the use of work-centre-specific tools. Although the responses of the suppliers also indicated the presence of this functionality, more intensive examination showed that this data is just recorded and not used in any functional way. So this incorrectness was not recognised by the data structure either.

7316 Based on the data structure it was shown a few times that certain functionality would be available, in contraction to the response of the supplier. During verification the response of the supplier was found to be too negative and, therefore, (unintentionally) detrimental for the assessment of the package.

7317 Redundant implementations were also found for example with respect to subcontracting. One of the BIS offers two implementations, each of which is capable of supporting the delivery of materials to the subcontractor. Each of the implementations do have certain different limitations. This demonstrates that although the responses to individual questions may be correct, the combination of functionalities may not always be possible!

7318 In a number of areas specialised BIS go into more depth than the standard questionnaire does. For example, in one of the BIS the functionality with respect to potency and grades is implemented in much more detail. Not only is this functionality not expressed in the questionnaire, a precise response can even result in an impression that the support is actually poor. For example, the question if potency per lot can be registered should be answered in the negative, because this functionality is present at a lower level (per lot-location) about which, however, no questions are asked. A correctly completed questionnaire leads to an incorrect conclusion. This leads to (not unjustifiable) manipulation of the responses by the supplier so that a correct impression of the software can be achieved.

7319 It follows from this that alternative philosophies in a BIS are not done justice in a standard questionnaire. A typical example of this is the availability of a variety of adequate implementations for the management of tools. These go further however than the simple question of whether tools are considered to be 'item', 'machine', or a separate entity type 'tool'. This is also true for two-dimensional resources for which a good 'work-around' emerged in one of the BIS.

7320 In one of the BIS it turned out to be possible to realise network structures for operations with a network of routings (although the terminology of the BIS concerned is somewhat different). This is accomplished using a number of functional 'tricks' in the software that are normally placed in data structures instead.

7321 Furthermore, one of the questions was associated with the assessment of purchasers. The supplier of the BIS responded positively to this question because the purchaser could be recorded in all of the relevant places in the data structure. The rest then has to do with user-specific reports. This could be easily verified based on the data structure. In comparisons of several BIS, some systems go much further, which leads to a complex data structure. The responses of both suppliers do not, however, differ.

7322 Finally, a number of differences were observed in the version of the BIS that was used in the response to the questionnaire and the version of the data structure that was assessed. Sometimes custom-made functions had been provided for a group of clients, and in the response to the questionnaire this customisation was regarded as a part of the standard package. There is something to be said for this, but it is important for the assessor to be aware of it. In addition, new functionality in some BIS is made available as soon as it has been developed, instead of saving the changes and extensions and implementing them together in, for example, one release per year. What is interesting here is the observation that these kinds of differences can evidently be detected on the basis of the data structure. Aside from the relevant information that it produces, it also in any case confirms the distinguishing power of data structures.

7323 Other kinds of differences observed were as follows:-

(a) deficiencies in the documented data structure; in particular, when the documentation has been set up by someone else than the software developers;

(b) the unintentional interchange of responses, something which is a risk with extensive questionnaires and multiple questions.

Analysis of the results in relation to the 'data-process' terminology

7324 The effectiveness of the assessment of data structures depends strongly on the type of question, and is thus associated with the 'data-process' terminology of the previous section. The results are analysed per type of question as follows:

DS - data singular

7325 The type 'data singular' inquires about the presence of (single) data types. These contain the typical characteristics of entity types that have to support processes and functions. One example is the question of whether scrap can be recorded as constant or variable amounts per 'bill of material relation'. This of course does not have to do with recording, but with the use of these attributes for the determination of material requirements.

7326 First of all, when searching for attribute types, problems with interpretation are encountered. In a data dictionary (or other system documentation) an attribute type has a name and sometimes an one-line description. It is not always possible to determine to what extent the same thing is meant by the developer and the questioner. For example, in one of the BIS studied, the following attribute types were found in the 'bill of material relation': 'fixed_qty_per', 'max_qty_per', 'min_qty_per', 'qty_per', 'per_qty', and 'qty'. In a BIS, where for example an item master file can contain hundreds of attributes, the introduction of an extensive user-oriented help function is not even an adequate solution for this interpretation problem.

7327 In spite of this risk of misinterpretation, the results show that a large number of attribute types can be recognised with a reasonable certainty. However, this result is strongly dependent on the quality of the data dictionary, the absence of tricks, and with it, the 'neatness' with which the BIS has been developed.

7328 The attribute types with a single connection to a specific functionality, such as potency or grade of lots, can generally be successfully recognised. This is in contrast to characteristics that can be realised in different ways in attribute types and can imply very different functionality. One example is the overlap of consecutive operations that can be expressed in time, units of end products, and percentages (this interpretation is usually 'hidden' in the software code). In the case the attribute type is a percentage, this can be expressed in relation to various other attribute types, and could also be considered to be either a minimum or a maximum. All this interpretation is 'hidden' in the software code and can not be deducted from the name of the attribute type.

7329 Secondly, the data-oriented approach can never achieve more than the determination that the attribute types concerned are either present or absent, without making any more statements about the associated functionality. The presence of the attribute type in question certainly does not imply that the associated functionality has also been implemented. Various attribute types have been encountered that were not actually used in the BIS, including an attribute type 'for future use'.

7330 However, *the absence of attribute types* be interpreted with reasonable certainty as meaning that the associated functionality can't have been implemented, since the necessary characteristics can't be recorded anywhere. This means that, in principle, the absence of associated functionality can be demonstrated on the basis of attribute types. The features that can be calculated from other attributes or combinations of attributes form an exception to this.

7331 In summary, the questions of type 'data singular' can be recognised from a data dictionary, in particular, if the quality of this dictionary is very high, as in the modern BIS. *The results of these questions should be interpreted with caution, but form a strong basis for a discussion of the content with the supplier* („I couldn't find this" or „in my opinion this isn't possible, how did you solve this?"). This need for caution is confirmed by the number of errors observed in the responses of the suppliers.

DM - data multiple

7332 The type 'data multiple' inquires about the presence of a collection of related attribute types, usually with a clear enough identity to be considered as an entity type. Such entity types can often be recognised one-to-one in the BIS under consideration, especially if this BIS has the same level of abstraction as that perceived by the questioner. Therefore, this is not the case for a question about 'tools' while the BIS recognises a higher level of abstraction, such as 'resource'.

7333 Whereas the interpretation of single attribute types usually has to be verified by the supplier, the presence of the entity type in question is virtually always self-evident. It does help in the interpretation to have an understanding of the associated attribute types, as these clarify the meaning of the entity type enormously.

7334 An exception is formed by the situations in which the development of the BIS was chosen in order not to normalise the given entity type, but 'to hide' it in attribute types. A few examples that were found include: recording 'tools' as 'items', where a single attribute type is distinctive; or alternative items that are recorded as a kind of repeating group within the entity type 'item'.

7335 *It can be concluded that this type of questions can be answered very straightforwardly on the basis of data structures.* This straightforwardness resulted in this research in the detection of a few errors. The fact that not to many errors were not found is probably due to the clarity of the formulation of the questions. It is relatively easy for suppliers to answer this kind of question. There is also a one-to-one relationship between the entity type being asked about and a data entry screen for the user.

DC - data complex

7336 The type 'data complex' asks about the presence of a complex data structure. This structure is often made up of several entity types whose relationships have a number of the specific characteristics. One example is the question whether tools can be recorded in relation to an end-product, an operation, a machine, and/or a customer.

7337 Just as with the questions of the type 'data multiple', these questions can also be simply and directly recognised in the data structure, again also especially when the same level of abstraction is used in both. If a certain BIS has a totally different philosophy and level of abstraction, then this philosophy needs to be thoroughly understood before interpreting the data structure. *The data structure is a (nearly essential) aid in coming to this understanding.* This philosophy should then be accommodated in the RDM.

7338 This type of questions can be more difficult to answer in some specific (exceptional) cases; namely, the situations in which either relationships are implicitly recorded in attribute types or functionality is realised in the software code and not in the data structure. An example of this are 'feed-ins of material in operations', where instead of proper relationship types between the 'bill of material relation' and the 'operation', this functionality can be hidden in sequence numbers that have to correspond in the two entity types. A typical example of functionality in software code is the realisation of 'a network of operations' by the implicit use of automatically generated dummy items that couple the output of the operations to the input of the next operations.

7339 The power of the data-oriented approach becomes most apparent with this type of question (refer again to the number of observed errors in figure 7.3). The fact that relatively more errors are found for this type than for questions of type 'data multiple' can be explained by the relative complexity of the functionality in question, and thus the difficulty in responding. Whereas the 'data multiple' questions usually have a one-to-one correspondence to data entry screens, the functionality focused on by the 'data complex' questions is often maintained implicitly by the software (consider for example the relationship between a work order and a sales order for customer order controlled manufacturing).

7340 *It can be concluded that these questions can virtually always be properly answered based on data structures.* The quality of the data structure delivered is the most important condition.

PP - process presence

7341 The type 'process presence' inquires about the support of a (often) extensive process for example 'the purchasing process' and 'the process of product configuration'. Such processes have an obvious impact on the data structure and contain in principle a complex of entity types and relationship types. *Based on data structures the questions can be easily answered.*

7342 However, this type of questions can be easily considered as trivial. Asking if the purchasing process is supported by a BIS is generally useless, since every BIS supports this process. Still, there are a large number of processes where, for example, the definition is somewhat less widely accepted, such as the support of the marketing process. For a number of suppliers, the presence of a mailing function is enough for them to claim that the marketing process can be supported.

7343 *The added value of the data structure is that at a glance it gives an impression of the amount of support.* In addition, it is possible to 'zoom in' on specific functionality within the process. However, if an assessor only wants to know whether a process can be supported, then the request for data structure is certainly out of proportion here. If this data structure has already been available, then these questions can of course easily be answered without any support from the supplier.

PSB - process step behaviour

7344 The type 'process step behaviour' asks about the behaviour of separate steps in the process; for example, whether the operation times are automatically adjusted based on the potency of the party of raw materials during the determination of the capacity requirements. This required behaviour is in no manner apparent from the data structure or any attribute types.

7345 The only roll that data structures can play here is in the disaffirming of this kind of functionality, in this example because the potency is not supported at all. *Therefore, based on data structures the **absence of behaviour** can in some cases be demonstrated.* The degree to which this is possible depends on the data elements required that would have to support this behaviour. This can vary from not even a single attribute type to a complex data structure.

7346 This was confirmed in this research by the large number of '?' in the verification of the answers on the presence of claimed functionality (see table 7.2), while the absence of functionality could be verified in more than two-thirds of the questions.

PSP - process step presence

7347 The type 'process step presence' asks about the support of a certain activity like a step in a process for example whether process quality control is recognised by the receipt of purchasing materials. The presence of such steps can be shown to be plausible by attribute types or even more complex data structures. In this example the presence of an attribute type 'number examined' besides 'number received' could indicate this.

7348 *Based on data structures the plausibility of the support of a given process step can be shown sometimes.* This is primarily because the performance of individual process steps will be recorded in attribute types for example in order to prevent an activity from being performed again. The interpretation of such attribute types is, however, risky and could suggest all sorts of process steps that don't actually exist.

7349 Comparable arguments hold with regard to the absence of process steps as discussed on the behaviour of process steps. Most of the process steps should be supported or recorded in one way or another in data elements. The absence of relevant data elements shows with high certainty that these steps can not be supported.

PSS - process step sequence

7350 The type 'process step sequence' inquires about the order in which activities or process steps are carried out. Although this type was not found in this research, it seems an obvious assumption that data structures give absolutely no information about this. *These questions cannot be answered on the basis of data structures.*

7351 These choices will only need to be recorded in for example parameter tables if the BIS supports alternative orderings in a process. The interpretation of parameter tables as part of the data structure is however an impossible task, and besides, generations of BIS are expected that will use other techniques to support this variability, like on the basis of work-flow management.

8 | Evaluation

8001 The testing in chapter 7 shows that many answers to questions about functionality can be verified on the basis of data structures. In only 17% of the cases does the data structure fail to produce an assessment regarding the functionality in question (the '?' in table 7.2). In addition, 'errors'[18] were found in more than 10% of the questions, even though the suppliers completed the questionnaires to the best of their knowledge and independent experts screened the responses.

8002 The characteristic results, advantages and disadvantages versus the required effort are evaluated and summarised from paragraphs 8100 and 8200. These results are presented first for data structures in general, and then for the RDM-method in particular. The characteristics of both approaches are summarised with respect to each other.

8003 From paragraph 8300, the RDM itself, which is the core of the RDM-method, is subsequently described.

8004 Finally, the practical applicability of the use of data structures in selection projects and evaluation processes of BIS is given starting at paragraph 8400.

The use of data structures

8100 This evaluation of the use of data structures involves consecutively:-
(a) the indirect character of the evaluation based on data structures;
(b) the possible conclusions that follow from the data structures;
(c) the flexibility in the use of data structures;
(d) the limitations of the data-oriented approach;
(e) the effort and expertise required with the data-oriented approach.

The indirect way of assessment

8101 When using data structures, it is important not to forget that it is an indirect form of assessment. The fact that a certain data structure is present does not necessarily mean that this functionality has been implemented in the BIS. It was argued in paragraph 7121 that in practice this indirectness does not cause any problems. An important argument that was given is that in practice developers will not invest a lot of effort in design in a data structure if they do not intend to use it in the further development.

8102 This means that the alteration of a data structure with modern fourth generation tools may be possible and easy, but that this in itself does not ensure that the complex, underlying functionality has been realised. For example, the addition of an attribute type 'potency' to a 'lot' is a relatively easy extension, but the consequence of this addition for the required functionality is much larger. Consider the calculations of the material requirements (and perhaps even capacity requirements), the material used, the confirmation of completion of production orders, and the many other functions. This is of course not true for single attribute types like text fields, but these are neither critical selection nor assessment criteria!

18 By 'error' is meant: differences in interpretation, differences in version, and the coincidental interchange of answers.

8103 The extension of the data structure with relationship types is even more complex. Here it is also true that the data structure itself is not the issue, but rather the underlying functionality and here particularly even the philosophy of the system. Consider, for example, the addition of a relationship type between sales orders and production orders. The creation of this one relationship type means that the planning, creation, and confirmation of completion of production are all directly coupled to the individual customer orders. Even more important, the control structure of the BIS was probably based on 'make-to-stock' and due to this extension the philosophy must be changed or enhanced with the 'assemble-to-order' concept. This has far-reaching consequences for virtually all of the planning and control functions within the BIS. Thesis 8.1 follows from this.

> *An adequate data structure is a necessary condition, but not a sufficient one, to ensure the functionality of a business information system.*

Thesis 8.1 - The necessity of data structures on functionality

Possible conclusions following from data structures

8104 Especially the indirectness of this approach, the assessment of the data structure, has important consequences for drawing conclusions. It can be derived from thesis 8.1 that the *absence* of functionality, and therefore the limitations of the BIS, can be proved on the basis of data structures; this is not the case for the *presence* of functionality.

8105 The data structure directly describes the philosophy of the BIS. This is primarily relevant for systems that are not based on the traditional MRP philosophy and APICS terminology, but rather on a quite different terminology, organisation, and characteristics (see for example paragraph 7319). Two typical examples are the specialist BIS, such as for the semi-process industry, and some typical customer order controlled systems. Also a number of systems on the low-end of the market have a special philosophy that better matches the pragmatism of the very small companies. *It is precisely the data structure that is a very effective tool (for an expert) in understanding these different philosophies.*

8106 *Furthermore, the data structure provides the context in which the functionality can operate*, thus which combinations of functionality are and are not possible. See among others the discussion regarding the test results in paragraph 7314. This discussion shows that the individual functions (and responses of the supplier) may not always be 'added'. Some functions may not work in certain situations or may exclude each other. In particular, questionnaires and demonstrations make no statement about this context (unless a specific question about a certain combination is posed, which of course cannot be done for all possible combinations).

8107 *Finally, data structures describe the implementations that have been realised in the BIS.* Usually several (alternative) solutions are available for a single problem. Each of these alternatives has its own features, advantages, and disadvantages. In principle, which of these solutions has been implemented in the BIS follows from the data structure. See, for example, paragraph 7303, where a group of machines is offered as a solution to dealing with alternative machines, however, if extensive functionality is demanded with respect to alternative machines, then this solution will be inadequate. This indicates that a detailed questionnaire must go into detail over a subject (through a series of questions) in order to get information comparable to that obtained based on data structures.

Flexibility in using data structures

8108 Good preparation of the questionnaire (or the business case or demonstration) is an elaborate activity that should not be underestimated. The objective is to cover all of the important aspects in one step. *An advantage of the use of data structures is that much more information can be gotten from them than could be gotten alone through answers to questions that have been devised and formulated beforehand.* The information from the data structure is not limited to what has been asked, but is rather, in principle, a complete description of the assessed BIS.

8109 *The property of being able to zoom in and out as needed in the data structure is related to this.* Data structures have an explicitly hierarchical structure of functional areas, groups of entity types and relationship types, individual entity types up to separate attribute types (depending on the method of documentation). This means that it is possible to zoom in to the aspect in which one is interested, for example, to recipes, their requirement generation, and specific features such as potency or grade. On the other hand, the evaluation of purchasing could be limited (zoomed out) to its relation to the other modules.

8110 *A distinctive (and essential) element in the evaluation of data structures is that one carries out the evaluation oneself, and not the supplier or some other third party.* In the last case a translation would again occur that would take away all of the advantages of the approach, which makes it equal to the normal manner of evaluation.

8111 *It also means that one draws (possibly preliminary) conclusions that form a good basis of discussion with the supplier.* The first conclusions based on the data structure in the testing carried out in this research also led to a very concrete discussion with the experts of the supplier's organisation. These were especially very enlightening regarding the differences and special characteristics that were found, see among others paragraph 7320 about networks of operations.

8112 In the analysis that led to the problem formulation (see chapter 3), the attainment of insight was found to be an important added value of data structures. Based on this evaluation, the **insight appears to be gained** in particular because data structures:-
(a) show the context and the possible and impossible combinations of functionality;
(b) make the philosophy of the BIS visible;
(c) make it possible to zoom in and out as needed in the description;
(d) contain much more information than a collection of criteria formulated in advance.

Limitations of the data-oriented approach

8113 As already indicated, data structures describe solutions that are implemented in a BIS. Choices must be made, just as with the design of a data structure in a specific company situation. Examples of choices are: the terminology used, what is considered an entity type (machine, tool, or resource) at a logical level, and which relationship types are selected to be modelled and which are not.

> *Data structures always describe a (design) result or solution, but they are not appropriate for the formulation of a problem or of the required functionality.*

Thesis 8.2 - The role of data structures for defining a solution and not a problem situation

8114 This means that information requirements or a desired business situation should not be described on the basis of data structures. Choices are made then that may be implemented differently in the assessed BIS and are perhaps even more suitable than what was designed.

8115 Some techniques (for example GRAI and IDEF) were presented in chapter 2 for the determination and description of a desired situation, such as the control structure and additional requirements. One of the steps in (software) development should be to model such a description as well as possible in (among others) a data structure. However, an evaluation of the existing data structures of a BIS should be based directly on the original description of the desired situation.

8116 Although data structures can play a meaningful role in the evaluation of a BIS, they remain just one aspect of the evaluation. A significant number of features can never be assessed using data structures. Obvious examples are supplier's criteria (see paragraph 2304) and the criteria regarding conditions of delivery (see paragraph 2306). This is also true for the many non-functional criteria that can be placed on a BIS, such as technology, performance, security, and documentation (see further paragraph 2305).

8117 There are also some functional features that cannot be assessed using data structures, as was confirmed by the testing in this research. *This primarily concerns the behaviour of the various process steps.* Especially functions at the level of decision support and (complex) algorithms are controlled on the basis of only a few parameters in the data structure. The behaviour to be assessed resides, however, in the software code. Partly due to the complexity of the software, it can really only be evaluated by testing in demonstrations or workshops.

8118 Apart from that, it is stated that standard BIS should not be tested on their absolute correctness, as self-developed systems are tested. Because such a BIS has already been implemented a number of times, in the context of the selection it should be assumed that the BIS no longer contains any technical or logical errors. Such errors should have been removed during the development and initial use of the system. Although this is perhaps not always the case, these kinds of errors will have been corrected in the following versions. Besides, carrying out this kind of evaluation - the real software testing - is an activity of a different magnitude and approach. The normal selection processes are not suitable for this.

8119 Finally, a warning to avoid the pitfall of filling out a questionnaire on the basis of data structures. The essence of this data-oriented approach is that the data structure serves as the answer to the functional requirement that have been stated. Both because of the limitations of the questionnaire itself as well as the answers in the form of 'Yes' and 'No', one should only perform the evaluation of the data structure oneself.

Effort and expertise

8120 Some important remarks with regard to the efficiency of this data-oriented approach in the evaluation of the data structures need to be made. The most important condition is of course the attainment of a qualitatively good description of the data structure. Given this condition, interpreting the data structure and making conclusions about it requires only limited effort. The testing itself only took a few hours for the most comprehensive data structure.

8121 The quality of the documentation of the data structures was found to have improved in recent years. This is mostly due to the software development tools used and the need of the suppliers themselves to understand the increasingly more complex BIS. However, there is still plenty of room for improvement in the quality of the documentation. In a number of cases it was necessary to first restructure the data structure and to draw a new diagram for it before it could be easily interpreted. This is primarily because the descriptions are made from the perspective of the development objectives of the BIS, and not from the objective of studying the BIS. For the improvement of the quality of this documentation, refer to paragraph 4200 onward, the same guidelines as are used for the description of the RDM.

8122 *It was also shown that - even if the data structure of a BIS is not available - considering a BIS and discussing it with experts in terms of data structures is an important aid in coming to an understanding of it and thus in assessing it.*

8123 Of course the most important condition for the interpretation of data structures is that the proper expertise is available. As already claimed, the person assessing the data structure should also interpret it. This person should therefore have knowledge and experience in areas such as the reading of data structures, and the assessment of different BIS (and different software philosophies). Furthermore, a background in management science is important for estimating the relevance of various functional aspects of a BIS.

8124 This leads to some demands on consultants and users who want to use the data-oriented approach, for example, in a selection process. Only when these demands for expertise are met can the previously discussed advantages of data structures be gained. The use of data structures, with their opportunities (in spite of their limitations), is a useful extension within the various management science education programs. However, it is essential that the correct usage be taught; see the preceding evaluation.

The use of the developed RDM-method

8200 In addition to the direct assessment of the data structure of each BIS, in chapter 4 a reference method (the RDM-method) was developed. The testing, which is carried out where possible on the basis of projections on the RDM, leads to this evaluation of the RDM-method.

8201 This evaluation examines the following in consecutive order:-

(a) the limitations of the RDM-method with respect to the direct assessment of the data structure of a BIS;
(b) the added value of the RDM-method;
(c) the effort required in the various phases of the RDM-method;
(d) the use of automated tools when applying the RDM-method.

Limitations of the RDM-method

8202 A strength of the direct assessment of the data structure of a BIS is the possibility of zooming in and out with this form of description. The RDM in fact recognises only one level of description; see chapters 5 and 6. This is a direct consequence of the elimination of as many details as possible. Although this generally strongly increases the readability and the clarity of the data structure, *in the RDM it is no longer possible to zoom in on details in specific areas*, such as the full description of entity types and the associated attribute types with their documented comments. The loss of details is inherent in the projection process of the original data structure onto the RDM.

8203 Another essential feature of the RDM-method is the uniformising of the data structures of different BIS in one model. The consequence is that *the specific terminology and aggregation levels of the original data structures are lost during the projection.* This can be partially prevented by including such specific structures in the RDM. The RDM can be extended in this manner with specific structures like 'Resource', 'Production Unit', and 'Production Model'. The creation of the extra redundancy in the RDM is in itself no problem; see the argumentation in paragraph 4234 and further (such an extension is, for that matter, an example of the continuous evolution of the RDM). Nevertheless, during the projection concessions will always have to be made regarding specific characteristics of the original data structure.

8204 The risk is that a strongly different philosophy of a BIS will remain hidden in the uniform terminology. It is still essential that this philosophy follows from the projection onto the RDM, but it should be realised that this can be 'hidden' in small differences of the projections: maybe just one relationship type or even only its cardinality. This means that the projections must be read and interpreted carefully, in contrast to the original data structure where this philosophy is much more explicitly visible.

Added value of the RDM-method

8205 An important advantage of the RDM-method is that the projections are much more readable than the original data structures are. As already indicated in paragraph 8121, in the current research it was often necessary to restructure the attained data structure. This was most true for the extremely extensive BIS that contain many thousands of entity types. The projection process is a manner of carrying out this restructuring, and at the same time, a simplification.

8206 Through the increased readability of the projections, fewer high demands are placed on the expertise required, although experience with the reading of data structures remains necessary. Still, the transferability, and with it the objectivity, is increased by the projection. The assessment or comparison of the projected data structures does not need to depend on just a single expert.

8207 In contrast to the usually totally different data structures, the projections are comparable. An example is shown in figure 6.12, where the projections of four different BIS are shown. It appears, for example, that only one of these projections does not recognise a relationship between the operations on the material requirements (the so-called feed-ins from the questionnaire in appendix C: 3.2.3). The comparability is both an important objective and an added value of the RDM-method.

8208 The RDM-method shows which aspects are supported and which are not, even when no questions about them are asked. Because of this, complicated structures that may not be needed are still visible. This gives a lot of insight into the abundance of functionality and complexity that one acquires. For the possible consequences of unnecessary functions, refer to paragraph 4415. This can not be confirmed by the testing because the questionnaire, as well as all the other approaches, is only directed at functionality which is present.

8209 The projection of the data structure of the BIS is a separate activity in the RDM that occurs prior to the assessment. The projection therefore had to be performed independent of any kind of assessment criteria. In a later stage, during the assessment or comparison, the same conclusions still have to be reached as at the time the projection was made. This places high demands on both the RDM and the projection process.

8210 With regards to the RDM, one should not be afraid to extend this during the projection (in fact this is an essential activity in the evolution of the RDM), so that the relevant features of the data structure of the BIS under consideration show up well. One should also be certain that the projection is performed correctly. This more or less forces the interpretation of the data structure to be done properly. Besides placing demands regarding experience on the person carrying out the projection, giving feedback about the projection to the supplier (or other expert on the BIS) is a good form of verification. When providing this feedback, attention should be given to especially the noticeable 'limitations' of the data structure. These are exactly the elements that will trigger the supplier and thus increase the quality of the projection.

8211 The RDM-method is an effective way of recording important features of a BIS. Especially during the selection process it should be possible to assess various BIS rather quickly. It is already known in advance that the direct, data-oriented approach can not do more than collect data structures. The RDM-method, on the other hand, offers the possibility of performing the interpretation and projection well in advance, as an investment in future selection and assessment situations. This investment is also a way of accumulating knowledge (in the form of projections onto the RDM) of characteristics of BIS.

8212 Finally, only a very fine distinction was found to have been made in BIS with respect to typologies of, for example, *'Items'* and *'Bills'*. However, as was shown with the RDM, this kind of typology can be modelled very well in data structures, using the mechanism of specialisation; see among others figure 6.13. This means that if the supporting typology has been worked out once for the projection of an existing BIS, then the projection onto the RDM is a good opportunity for recording this result.

Effort in the different phases of the RDM-method

8213 The RDM-method distinguishes several activities:-
(a) the development of the RDM;
(b) the interpretation and projection of data structures onto the RDM;
(c) the comparison and assessment of projections with respect to requirements;
(d) the evolution (development) of the RDM; this is discussed further in the following section, 'The use of the RDM'.

8214 The development of a good RDM is not trivial, not only because of the size, but primarily because of the required quality. The effort spent on development is in principle only needed once, with the exception of that expended for the continuous evolution of the RDM, which is an important element in the use of the RDM-method.

8215 The projection is also a one-time activity that must be performed for every BIS being assessed. This has the advantage that the discussion with and feedback from the supplier is also only a one-time activity. The use of a projection more than once means a savings in effort compared with the direct assessment of data structures, where a verification must be performed each assessment. However, the projections will require maintenance because of:-
(a) new versions of the BIS. In practice these new versions are extensions of older ones, so that maintenance is reasonably limited;
(b) extension of the RDM in the scope of the development (an evolution).

8216 The final assessment of the projection can be performed at a different time and by a different person. Assuming the proper expertise, this assessment requires only a very limited amount of effort.

8217 In summary, the most important investment is the development of a good RDM. In combination with the one-time projection of the relevant BIS, the RDM-method is only profitable when used regularly. Especially the specialised consultancy firms that regularly carry out BIS selections can profit from the RDM-method (perhaps, thinking further along these lines, a kind of certification could be considered for which the RDM would serve as the norm for BIS).

Tools for supporting the RDM-method

8218 It seems worthwhile to support the RDM-method with automated tools. These kinds of tools should be able, among other things, to record the structure of the RDM as well as the projections of the different BIS onto the RDM.

8219 Many tools are available for recording data structures, most of which form a part of a CASE tool. A special requirement for the storage of an RDM is that it must be possible to disable the normal consistency-checking for data structures. The RDM does not need to be consistent (refer to paragraph 4234). In particular, alternative structures should be able to be stored within a single version of the RDM.

8220 Besides the structure of the RDM, a comprehensive explanation should also be able to be stored in the tool. This is an important part of the RDM for lowering the level of expertise needed for the interpretation. The explanation covers entity types, relationship types, substructures, and the various subjects (as, for example, were dealt with in the description of the RDM in chapters 5 and 6). A coherent presentation of this information should of course be provided.

8221 Some secondary demands that may be placed on a tool are:-
(a) flexible searching on subjects and keywords;
(b) simple manoeuvring between the different (sub)structures;
(c) selective (and clear) presentation of the many relationships between entity types, depending on the substructure concerned.

8222 The RDM-method should be supported by recording not only the RDM itself, but also the projections onto the RDM. Storing these projections separately will not be a problem with most tools; however, retention of the relationship between projections and the RDM is, as far as is known, not possible with any of the tools. If this relationship is not stored, then all of the changes in the RDM will have to be changed manually in each of the projections. Because this is impractical, additional development of for example an existing tool will be necessary for supporting the RDM-method.

8223 The Aris-Toolset [IDS94] might be a good starting point for the development of a tool. The advantages of this toolset are:-
(a) the comparison mechanism for comparing and analysing models with different origins;
(b) the integration with other types of models such as a process model and a function model. Given the limitations of the data-oriented approach, it may be possible to record some of the additional characteristics in this toolset. Judging the adequacy of this is outside the scope of this research, but the Aris-Toolset at any rate presumes to claim that it covers all of the relevant aspects of BIS;
(c) the effort that is expended already for the description of a number of BIS as well as reference models with Aris-Toolset (this primarily concerns process models and not data models).

8224 The big difference between the Aris philosophy and the RDM-method is that Aris records all BIS in their own terminology and that furthermore the basic assumptions and principles are fundamentally different. Furthermore Aris is based on industry models as a reference, which differs from the assumption of the RDM, and the emphasis in Aris is on process models and certainly not on data models. Nevertheless the integration of the Aris philosophy, the toolset and RDM-method can be very useful.

The use of the Reference Data Model (RDM)

8300 The core of the RDM-method is the RDM itself. Within the scope of this research an RDM was developed for the areas of 'Purchasing' and 'State-independent Manufacturing Data'. The RDM is evaluated here with based on the experience during the development, but also based on the results of the testing.

8301 This evaluation examines the following consecutively:-
(a) the quality of the RDM, which determines the success of the RDM-method;
(b) the RDM as growing model;
(c) the value of the RDM itself, independent of the RDM-method.

Quality of the RDM

8302 The success of the RDM-method is determined by the quality of the RDM. The *discriminating power* is of particular importance for this quality. For this reason several different data structures are projected onto the RDM for 'State-independent Manufacturing Data'. This discriminating power is, for example, illustrated in figure 6.12, where four BIS are projected onto a single basic structure. *In this figure it can be seen that the differences largely concern the relationships between entity types, and not the presence or absence of entity types themselves.* The discriminating character of relationship types seems to be typical of the most differences in the projections. Only if a function area is not supported at all, such as for instance a product configurator, the related entity types are absent.

8303 It can also be shown from this illustration that it is possible to develop a single RDM in which fundamentally different BIS and philosophies (bills of material and routings versus the production model) can be accommodated. The projection reflects the original data structure. This property is essential for being able to compare the various BIS.

8304 The RDM should provide insight into *alternative solutions* and implementations in the different BIS. Several examples of this can be found in chapters 5 and 6, including:-
(a) co- and by-products, see figures 6.40 and 6.41. For both of these, the most extensive structure was determined, with actual implementations being a subset of these structures. It is shown how the different implementation for 'by-products' are related to each other (for example the negative material requirement in the 'bill of material' versus the separate entity type 'by-product'). It can also be derived from this structure that the implementation of 'by-products' is a specialisation of 'co-products', and is therefore implicitly covered by an implementation of the latter;
(b) packaging of purchased materials, see figure 5.6, where three (alternative) implementations are shown that are included in the RDM. Each of these alternatives was found to be present in the BIS examined; however, one BIS will never implement more than one of these alternatives, since it would introduce unnecessary redundancy.

8305 The RDM should be useful in distinguishing what is and is not important in the assessment of BIS. In principle, the development of the RDM is directed at this objective. The RDM was altered in the exceptional areas where this was found not to be the case during the testing. Based on the expertise acquired during the development of the RDM regarding the important and characteristic features of BIS, the formulation of many questions in the questionnaire (appendix C) was discussed during the testing and then formulated more concrete. Examples are:-

(a) the question simply about multiple 'bills of material' (appendix C: 3.2.1). However, a real difference was found to exist between these different 'bills' (see the RDM, paragraph 6231), namely:-
   (i) for different purposes (planning, costing, production);
   (ii) as an alternative if certain components are temporarily unavailable;
   (iii) in relation to the retention of a history and the version management;

   if this question is not made more specific, then a 'Yes' from the supplier has not much value;
(b) the question about storage tanks as limiting resource (appendix C: 4.1.7). The question formulation does not take into account that such tanks have a state-dependent availability, namely at the start of production the tank already contains something. This implies a link with the stock location system and the tank as a resource. Such complex but essential functionality of tanks is not demonstrated by the question;
(c) the question about alternative item-supplier combinations (appendix C: 6.1.21). Regardless of the attempts to formulate this question well, the essence of the question was found to be unclear in the communication with the supplier. The importance and the kinds of alternatives becomes directly obvious when drawing the data structures as in the RDM. A sound formulation of the question would actually have to be made up of three questions (which would make the questionnaire even longer).

> *The structures in the RDM are potential knock-out criteria in the selection of business information systems.*

Thesis 8.3 - The importance of structures in the Reference Data Model (RDM)

8306 From the evaluation it appears that especially the RDM has a large influence on a good understanding of a subject and, consequently, on posing the right questions and having the right discussion with the supplier.

RDM as a growing model

8307 Throughout this research and in particular during the testing it was confirmed that the RDM must be considered a growing model. When a new BIS or another problem areas is examined, the RDM will evolve. *In essence the RDM is just as dynamic as the (underlying) BIS and as the problem areas it describes.* If one is aware of the dynamic character of the RDM then this is not a problem when using it.

8308 These dynamics, however, do not say anything about the usefulness of the RDM in relation to its breadth (and level of detail). The dynamics are caused by the introduction of new BIS and new concepts that apparently were not yet important in earlier evaluations. For this reason, using the RDM is the best way of improving and further developing it.

8309 The use of the RDM within the RDM-method is directed at the projection and assessment of BIS. Within this application area, and in particular as part of the selection process, explicit attention should be given to the evolution of the RDM. This growth occurs through:-

(a) the projection and assessment of new BIS in which alternative solutions emerge that must be added to the RDM;
(b) the discovery of new philosophies as a result of considering new BIS. This could involve typical branch or production characteristics, such as perhaps from the textile industry if it involves colours or sizing, or from the sheet processing industry where optimising scrap plays an important role;
(c) the determination of important assessment criteria in a specific company situation that is not yet covered by the RDM. Although the thesis is that a majority of the important functional assessment criteria is expressed in terms of data structures, criteria can also be involved that do not follow from the data structure. It is proposed to add this limited number of characteristics to the descriptive part of RDM. This possibility is examined in paragraph 8313 and further, 'Stand-alone value of the RDM';
(d) the discovery of weaknesses and limitations of the RDM in general use. If these concern the functionality then they should be explicitly stated and included in the RDM in order to increase the quality.

This means that the RDM should not only be applied in a straightforward way, but that one should be alert to unknown limitations of the RDM.

8310 The above-mentioned extensions can theoretically lead to an unmanageable growth in the RDM, especially if there are a large number of BIS, philosophies, and implementation alternatives. Most of the many BIS now available on the market are already covered by the RDM. Most extensions can be expected from a few innovative suppliers who base their development on scientific research, not more than a small, exclusive group of suppliers.

8311 The number of philosophies in BIS is even smaller than the number of innovative suppliers. Most of the BIS just seem to follow in the development of a sort of 'best practice' functionality. Compare for example the development of word processors, where only occasionally a new concept is introduced and where it is nearly standard practice for the competing software programs to copy each others features. Partly because of this trend toward hybrid systems, especially the controlling philosophies of the BIS seem to be converging. An example is the MRP philosophy that is supported by most BIS. The exceptions are focused on totally different forms of operation and controlling.

8312 Theoretically, there appear to be many alternative solutions. However, when standard BIS are involved, all the suppliers try to increase their standardisation so that they cover as much of their market as possible. This leads to a kind of 'best practice' implementations and converges to just a small number of genuine alternatives.

Stand-alone value of the RDM

8313 Beside the RDM forming the basis of the assessment of BIS, the RDM also has an intrinsic value. Because of the manner in which the RDM is developed and evolves, it more or less describes the entire state of the current development of systems. The collection of expertise and experience can be used among other things for:-

(a) education in the area of information systems, for example, 'how are BIS constructed', 'which typologies and characteristics are recognised', and 'applications and limitations of BIS';

(b) development of new information systems, especially for in-house development. Given the possibilities of modern development tools, it might become profitable again for companies to develop company-wide information systems themselves. The largest bottle neck is however determining all the system specifications in detail and to develop them from scratch. The RDM can be used as a start for this.

8314 Because the RDM is developed on the basis of the data structures of BIS, several aspects are (initially) omitted. Although data structures make an important part of the possible and impossible functionality visible, part of the functionality remains hidden. Therefore it is recommended that the RDM be extended with important functional characteristics that are not covered by data structures. A good example is 'potency', which is concealed in a few places in the data structure as just an attribute type, but certainly plays an important part in a number of situations in the semi-process industry.

8315 The limitations of the RDM primarily concern:-

(a) the detailed process steps that are supported (the type of questions in the categories PSP and PSS, see paragraph 7211). This means that the RDM should be extended to include process flow descriptions. Several suppliers are already working on such models. This limitation is an argument for an integration of the process-oriented and data-oriented approaches;

(b) the behaviour of the more complex process steps (the type of questions in the category PSB). Primarily formal languages should be used here. In fact, this is going in the direction of the object-oriented approach, where the behaviour of each object, its interaction with the outside world, is described.

8316 This last step, where object orientation would be added to the RDM, is a drastic one and means that the RDM is viewed from a different perspective. This is discussed further in chapter 9.

How two approaches can strengthen each other

8400 Software selections and the 'best practice' approach was discussed in detail in chapter 2. The selection is one of the most important moments during which BIS are assessed. Here the manner in which the data-oriented approach can be used and integrated in this 'best practice' method is examined.

8401 However, before examining the integration of the use of data structures, it must be emphasised that data structures describe design results or solutions and are not intended or appropriate for the formulation of a problem. If functional requirements were to be expressed in terms of data structures, then the alternative solutions (in data structures) would automatically be excluded. The assessment process is therefore a balance between the abstract requirements (the problem formulation) and specific implementations (the solution formulation). This distinction is very important in the selection of BIS, even when the tradition of using a questionnaire is upheld.

8402 If, however, the only objective of the selection process were to be the choice of an appropriate BIS, then the application of data structures would cost more effort than it would save. However, this is also true for many other activities that are normally included in the selection. Such crude choice processes, for example for smaller organisations, usually limit themselves to the availability of the required modules.

8403 The objective of the usual selection is broader than just the right choice, namely also:-

(a) the appropriate supplier as a future partner;

(b) the acquisition of enough knowledge of the BIS to be able to delineate an achievable implementation;

(c) the formulation of a rough plan for this implementation;

(d) the creation of support in the organisation for both the choice and its intended implementation;

(e) the negotiation of an optimal contract.

8404 In chapter 2 the selection process was divided into four phases: the pre-selection, the detailed selection, the test & demonstration, and the completion & negotiation. These phases will be treated separately below, except for the pre-selection (the creation of a long list of potential candidates), where data structures are virtually without added value.

Detailed selection

8405 In the detailed selection the 'knock-out criteria', or the 'important requirements', are determined. Normally the suppliers indicate to what degree the BIS that they deliver meets these requirements. Some specialised consultancy firms keep ahead on this by keeping these kinds of characteristics of BIS in a database.

8406 On the basis of data structures of BIS many (most) of these features can be directly ascertained without the need for feedback to the supplier. An expert performs the translation of the problem formulation into acceptable solutions in the form of data structures. The data structure in fact takes the place of the above-mentioned database of BIS characteristics.

8407 It goes without saying that the availability of the data structures of the software under consideration is a condition for use. This can especially be a problem in this early phase of the selection process, when a relatively large number of systems are still candidates. This means that an investment must be made beforehand in order to get this documentation. This investment is comparable to (or maybe less than) that for setting up the above-mentioned database.

8408 With respect to approaching all the suppliers on the long list, the use of data structures means a sizeable investment. However, it has the advantage that, as when setting up a database with BIS characteristics, a reliable and objective short list can be very quickly determined. Such an investment is only profitable if it is used repeatedly.

8409 The collection of data structures results in much more (important) information, and more than can be thought of beforehand, than a database with BIS characteristics does. In order for such a database to be functional, most possible requirements should be added to it beforehand. This means that first a nearly exhaustive list with requirements must be created and maintained. When data structures are used, new requirements can be tested on the data structure whenever they appear.

Test & demonstration

8410 Demonstrations of the short-listed candidates (based on a specific company case) is important for the testing the BIS. Naturally, an assessment of the data structures alone should not replace this essential step, which is also very important for creating support in an organisation.

8411 The added value of data structures is the identification of potential weaknesses of the BIS prior to the demonstration. Because data structures are directed at the potential weaknesses and therefore feedback to the supplier is so important, these weaknesses should be inventoried in advance. The demonstration is the proper time to try to make these weaknesses explicit or to negate them.

8412 This advance information can also help to describe as specifically as possible the company case that must be demonstrated. The most important thing, however, is to critically investigate the subjects during the case demonstration that are poorly supported, or not supported at all, by the data structure. The trick is to pin these critical questions to demonstrations of different processes and to translate them into users terms. Jargon, for example, in the area of data structures, must be emphatically avoided so that the users present can draw their own conclusions from the supplier's reaction and the demonstration.

8413 This sort of preparation requires an expert and must not be underestimated. In principle this expert function could be filled by someone from one's own organisation, but the experience required with the interpretation of data structures of BIS is essential. The final result of using data structures here will not save any time but will ensure an improvement in the quality of the case demonstrations that are more focused on the weaknesses of the system. This will be particularly evident in a more concrete, content-oriented, evaluation of the different demonstrations.

Completion & negotiation

8414 The concluding phase is made up of contract negotiations, sometimes preceded by a comprehensive workshop in which the basic choice is studied in far more detail. An important element in this phase is the preparation of the actual implementation. Data structures can be used in several places in this preparation, namely:-

(a) for coming to an understanding of the system as part of the training of the key users in the workshop. Insight must be gained into the typical concepts and the relationships between the various entity types of the BIS;

(b) for the estimation and specification of customisation of parts of the BIS that do not meet the requirements. Although this should in principle be done by the supplier, it is certainly worthwhile to form an impression of the attainability, the extent, and the complexity of such customisation;

(c) for the estimation of changes of the working practices in areas where the BIS does not completely meet the requirements;

(d) for the specification of a plan for inputting or converting all the data. The data should be entered in a logical and efficient order. Each BIS has its own preferred order in which all the data must be entered. The supplier can also play a significant role here.

8415 In this phase the choice for a given BIS has in principle already been made. For the use of data structures this means that one should focus on the actual data structure of the BIS itself and that projections onto the RDM are of no additional value. From this phase onward one of the most important activities is to become as knowledgeable as possible about the ins and outs of the BIS.

8416 The data structures can be applied selectively in this phase, depending primarily on the specific need and situation (for example, where customisation is necessary). Depending on the extent of the preparations for implementation during this concluding (selection) phase, some activities will continue on into the implementation itself. The application of data structures will be shifted somewhat from the expert to the key users when this happens. Especially the key users will need to understand the BIS being implemented in detail.

Evaluation summary

8417 In summary, the essential conditions for application are:-
(a) the availability of well-documented data structures of the BIS to be assessed;
(b) the presence of the expertise and experience in the proper interpretation of data structures.

8418 The level of investment required and the added value of the use of data structures are shown below for the four phases of the selection process. Refer to chapter 2, figure 2.1 for a description of these phases.

	Approach	Added value	Investment
Pre-selection		none	not applicable
Detailed selection	RDM or direct	only when frequently reused, instead of a database of software	high
Test & demonstration	RDM or direct	large, preparation of critical points for the case demonstration and testing	average
Conclusion & negotiations	direct	selective application, especially for acquiring knowledge of details	low

Table 8.1 - Investment and added value of using data structures in selection processes

8419 The application of the RDM-method as differentiation of the general assessment of data structures is only profitable when software selections are carried out very frequently. The investment needed for the direct assessment of the data structures of BIS is significantly lower.

9 Technological developments

9001 The added value of data structures in the assessment of BIS follows from the preceding evaluation. This is, however, based on the current state of the technological development. In this chapter the possible influence of these developments on the role of data structures as an assessment instrument is discussed.

9002 The technology on which the current generation of BIS is based is developing rapidly. In chapter 1 several important developments were outlined. A number of these are expected to have consequences for the role of data structures, especially the development of:-
(a) case tools and case templates;
(b) object-oriented business information systems.

9003 Although it is difficult to predict the success and the speed of these developments, some consensus does exist about the direction. This is apparent from, among other things, the sizeable investments being made by the software suppliers, especially in the area of object orientation.

Development of case tools and case templates

9100 Case tools have already existed for many years and their origin lays in the almost natural step following the development of procedural programming languages and design tools (see also chapter 1). However, the current generation of case tools do not yet seem to be competitive with the supply of standard BIS, at least with respect to commercial, distribution, and production companies. Even so, it is almost inevitable that such case tools will become increasingly better and will eventually be competitive in the development cycle of analysis, design, and generation of complex BIS.

9101 Custom systems design is, aside from being an advantage, also a fundamental bottleneck in the case approach. Primarily driven by a company situation and working practice, as well as a limited frame of reference, the complete design should arise from its own strength. This requires a lot of effort and the risk remains that non-optimal choices, or even design errors, will be made; consider, for example, the complexity of the MRP calculation that must be correctly designed and realised. Although a prototyping approach can help avoid such problems, prototyping does not work for the total functional breadth of the competitive standard BIS.

9102 The need arises in this development for a kind of basic design or 'best practice' design for different types of companies. If this were available, the design process could concentrate more on removing functions. The basic design would only need to be expanded in very enterprise-specific areas. Although the removing of functions seems simple, it should not be forgotten that it is difficult to make the result as small as possible (thus without any surplus).

9103 In the case tools generation intended here, the subject of an enterprise-wide data structure is central[19]. As argued in this research, the data structure contains a large number of design decisions. Based on this functions are designed and developed. Nevertheless it must be very clear which functions are required before designing the data structure. Besides, in the case approach more and more functionality will shift to the database, namely as 'triggers' and 'stored procedures'. These 'stored procedures' will be linked by work-flow techniques.

9104 Of course, in this kind of custom development the assessment of data structures is not relevant. The RDM-method as an assessment method does not have any added value. For the RDM itself it is a different situation. Although the RDM should not be used directly as a data structure to develop a BIS (see the typical characteristics of the RDM in chapter 4), it is a very good starting point to for this development. By stripping as many entity types and relationship types as possible, a consistent and implementable data structure can be quickly deduced. In fact the RDM is used here as a sort of 'template' from this development can be started.

9105 Besides being a starting point for the development of a specific business information system, the RDM can also be used for the development or deduction of case templates for special industry sectors, controlling structures or production processes. These are, according the definition of the RDM, a subset of this overall model.

9106 The greater the application of case tools, in a market that currently is controlled by standard software, the greater the role for the RDM. The further broadening of the RDM to (all) other functional areas will then be necessary. The result could be a commercially brought to the market as a basic model for this group of developers and future users.

9107 However, this development of case tools can not be seen independently from of the development of object orientation described hereafter. The success of both developments will certainly influence each other.

Development of object-oriented business information systems

9200 The development of case tools, as described before, can be seen as an extension of the current generation of development tools. The introduction of object orientation is much more revolutionary, see also the discussion in chapter 1[20]. Although the principle has been known for years (since the end of the 'sixties), object orientation is currently the most important issue for the large software-development organisations. Object orientation is seen by many experts as *the* trend in the area of information systems.

9201 The tools that are developed for this will often be categorised as case tools. However, an important difference with these 'traditional' case tools and their successors is the emphasis on the assembly of completely independent objects.

9202 The object approach is fundamentally different from the traditional procedural design and development approach. This necessarily has an influence on data structures and, therefore, on their role and place in the assessment as well as in software selection.

---

19 A different set of case tools is based on object orientation. Because of the fundamentally different approach, this also has a different influence on the role of data structures. This approach, possibly supported by specific case tools, is discussed from paragraph 9200 onwards.

20 In chapter 1 the most important concepts with respect to object orientation are stated. The possible development of a standardised framework for object assembly, the combination of objects from different suppliers, is also discussed.

9203 In view of the fact that object-oriented BIS are at an early stage of development, it is impossible to estimate all the consequences with respect to data structures as an assessment instrument. It is possible that this development will form a threat to the data-oriented approach, because all data are encapsulated in objects, and the assessment of individual objects will be performed based on other criteria such as behaviour.

9204 A more systematic approach to data structures and their importance in an object-oriented environment results in the following sub-issues:-
(a) the central enterprise-wide database;
(b) the structure between objects;
(c) the internal data structure of messages;
(d) the internal data structure of objects.

9205 It is apparent from this classification that the data structures can be divided into local, independent data structures next to the single, central structure.

Central enterprise-wide database

9206 It seems plausible that the object-oriented BIS will continue from the perspective of a (usually relational) enterprise-wide database, although this may not be an optimal form of object orientation where all data (and functionality) is encapsulated in objects. This central database contains only the data that is important for several objects. Examples could be: the current and future stock position that must be checked or updated from many perspectives; or maybe the chart of accounts in the central general ledger.

9207 The arguments used for this theoretical, perhaps impure, application of object orientation are:-
(a) *the availability of the enterprise-wide data*, that must be accessible in many places for various applications, including those which are not object-oriented;
(b) *the efficiency of retrieval of data*, which is extremely high for the optimised technology of relational database management systems;
(c) *not wanting to lose the knowledge and experience acquired*, and advantages of such enterprise-wide databases.

9208 A consequence is that such a central structure is completely stripped of all local details, leaving a very transparent enterprise model. Data structures as assessment instruments will retain their advantages for the purpose of choosing between such object-oriented BIS; the approach remains identical to the current generation of BIS, which are also based on such central databases. One possible advantage is even that the assessment or the projection will be easier because the central structures have been stripped of all unnecessary (local) details.

Structure between objects

9209 All object classes in a BIS are related to each other in an object structure, which is in principle implemented in an object-oriented database. Therefore, for example, a relationship could exist between the object classes 'sales order' and 'purchase order' in a commercial environment.

9210 One problem is that the choice of object classes in BIS is not clearly defined. Therefore, sales orders, contracts, and production orders can all be considered object classes, as can clients, suppliers, or batches of goods. However, other choices are also possible. In this development stage the broad 'best practice' experience is with regard to the choice and scope of object definitions in BIS lacking.

9211 The structure between object classes is often modelled with entity-relationship diagrams and resembles the data structures in this research. The object classes are, however, certainly not equal to entity types, but embrace a more complex data structure. In addition, the behaviour of the object is, naturally, also part of the definition of object classes, but other techniques are used for this description.

9212 Because object classes themselves contain an (complex) internal data structure, the structure between them is simpler than the entire data structure of the current generation of BIS. This does not make the assessment of this structure any less necessary, but it will result in less information and detailed insight. At this moment it is difficult to estimate to what extent this object structure will remain a selection criterion.

9213 The structure between objects is an important aspect of a framework for object assembly. The RDM could be an impulse for the development of this framework. This, in itself is, however not enough; standardised messages also need to be a part of the framework definition.

Internal data structure of messages

9214 Messages provide for the information exchange between objects. These messages are collections of data with their own internal structure. Although these messages are sometimes simple in nature, their structure is essential, certainly in the framework of standardisation. Such messages can be compared to message definitions that have been placed in EDI environments; here too, the standardisation is the most important issue.

9215 The role of data structures as an assessment tool is only relevant in the object assembly situation, and not in the selection of integrated object-oriented BIS. With object assembly the objects of different sources and releases should be assessed on their ability to work together. The data structures of the messages from object classes should be viewed with respect to their fit with the methods of the 'receiving' object classes.

9216 The standardisation required for this could of course be enforced by an organisation like OAG (this international committee is working on such standard messages under the name 'Business Object Documents') or the OMG (Object Management Group). If such a standard (and only one!) is completely accepted, then this assessment function would disappear.

Internal data structure of objects

9217 Besides the fact that objects show behaviour, which is principally triggered by messages, objects should also retain their (internal) status or state throughout their life cycles. For example, for the object class 'production order' a large number of order details are recorded, for example, the progress of the production, and which operations are complete. This data stores the latest status until, in this example, a message is received indicating that the following operation is reported as completed.

9218 This internal data can sometimes (depending on the definition of the object) be requested by the outside world via messages. In this example a message could inquire about progress with an object 'production order'.

9219 The role of data structures as an assessment instrument seems to be most interesting in the above-mentioned situation of object assembly, namely to assess the alternative objects from different suppliers. An important condition is the creation of libraries of interchangeable object classes and thus the required framework.

9220 The internal data structure of such objects could also become sufficiently understandable that the behaviour and the methods, would be more important. In this research the behaviour was more or less assumed during the identification of data structures. The attention could then, however, shift to the assessment of object behaviour.

Conclusion about the role of data structures in object orientation

9221 The above-mentioned developments are not anywhere near to reaching maturity. Especially the aspect of standardisation is very complex, not only functionally, but also because of the interests of the parties involved. Despite these uncertainties an attempt has been made to draw a conclusion about the future value that data structures could have as an assessment instrument.

9222 In fact, the possible developments in the area of object orientation can be divided into two future situations:-

(a) *a standard framework is developed (and widely accepted) on the basis of which object assembly is really possible.* This framework will be largely described in terms of data structures. In this situation it can be assumed that object classes, by definition, fit into this framework. The role of data structures as assessment instrument will be limited to the internal structure of object classes: which of the available object classes is the most appropriate for the stated requirement? It can not be predicted here whether the data structure will give adequate or even abundant insight, or whether it instead will not be important at all;

(b) *no framework evokes for object suppliers to conform to.* Perhaps a rough guideline will be developed, but the large number of variations will lead to (intensive) verification of the ability of object classes to work together. Then, it can be expected that the importance of data structures as an assessment instrument will increase, not only in order to assess the internal structure of object classes, but also to assess whether the relations between them and the communications via messages are appropriate.

9223 In any case, there is the perception that object orientation can cause a revolution in many areas of software development and software selection, as well as in the implementation of BIS.

Recommendations for further study and use

10001 The evaluation of data structures as assessment instrument was worked out in chapter 8 and was also based on a quantitative testing. An important conclusion is that insight that is gained is further clarified through the possibility to:-
(a) give a context in which functionality can work;
(b) make the chosen philosophy of the system visible;
(c) zoom in and out in this structured form of description;
(d) provide (much) more information than had been thought of in advance or inquired about.
Refer to chapter 8 for more explanation and additional conclusions.

10002 Of course, the limitations of data structures in this role need to be realised. However, given the knowledge of both the strengths and weaknesses, this can be used to focus the (assessment) energy on the right aspects.

10003 Based on the results of this evaluation, the broadening of this application of data structures is discussed here, namely completely different functions and branches than were dealt with in the current research can expect comparable advantages. Attention is then given to the further development of the RDM, which has an fundamental role in the assessment. Finally, the aspect of education is discussed as a necessary condition for dealing with data structures.

Broadening the application of data structures

10100 It can be expected over the complete functional breadth of BIS that the manner of assessment as well as the results will be comparable. This means that the added value as assessment instrument is primarily present in areas where:-
(a) the functionality is complex because of its details; examples are electronic planning tools or product configurators. This means, among other things, the assessment of specialist often stand alone software packages;
(b) the functionality is complex because of its breadth; examples are production control or trade & distribution. This means, among other things, the assessment of integrated information systems.

10101 For employing data structures in other functional areas in a comparable way, the following features in particular should be paid attention to:-
(a) an adequate number of registrational elements must be present. This is probably not the case in a number of specialist software programs that lean heavily on 'operations research' techniques, such as, for example, nesting software or routing software;
(b) some homogeneity with the functional area must exist, without this having been completely standardised. For example, an administration for accounts payable, accounts receivable, and a general ledger is probably so standardised (partly due to governmental rules) that data structures of standard packages are virtually non-discriminating.

10102 This research is primarily directed at functional areas that are important in 'industrial production' and 'trade'. Other specific branches where it can be expected that data structures could have a large added value are:-

(a) *the insurance sector.* The registrational character is large here, with a broad complexity and variety of products and insurance policies. Furthermore, the level of standardisation is still limited by the rapid renewal of insurance products and the strategic importance of information technology in this sector. Also the flows of processing mutations are very important in this sector. Note that data structures make no statements about this;

(b) *the transportation sector.* Registration is a primary function here as well. This is generally complex because of the variety of parties (shippers, transporters, and shipping- and forwarding-agents), the different transportation modalities all with their own characteristics, and the international character of this sector. A lot is invested in standardisation, but until now is has been with only limited success. This is partly caused by the enormous competition and the abundance of (very) small companies which is, for example, in contrast to industrial production.

Further development of the RDM

10200 In this research the RDM was worked out for two functional areas. Given the results of the evaluation, the RDM could be extended further. This means primarily the broadening of the RDM to other functional areas such as stock control and sales. The effort required to achieve this and the result of this effort can be expected to be comparable to those of the current research. Parallel to this, data structures of the important BIS should be collected and projected, so that not only the breadth, but also the quality of the RDM increases. This parallel development seems essential in order to be able to guarantee a qualitatively useful RDM.

10201 The experience gained with this research teaches that this sizeable effort can, however, only be profitable for specialised consultancy firms. Especially a consortium of interested parties might be in a position to make use of such an investment.

10202 It seems extremely worthwhile to introduce the RDM-method in the development of the Aris-Toolset. The support with an automated tool can have the following advantages:-

(a) the maintenance of the RDM and of the projections made onto it, in particular, throughout the evolution of the RDM;

(b) the support of the projection process;

(c) keeping the RDM accessible as the size of the RDM increases.

10203 The Aris-Toolset now offers a series of functions related to the demands placed on automated tools by the RDM-method. For this reason a combination is an obvious choice.

10204 Another possible development is the extension of the RDM to other (not data-oriented) aspects. Some suggestions are made about this in the evaluation presented in chapter 8, though without having tested the feasibility. The most important extension concerns the addition of behaviour, which strongly resembles a step in the direction of object orientation. With this the RDM develops in the direction of the framework of object classes. In fact this is a very useful development, but then it should be elevated to the primary objective. A collaboration between a number of prominent suppliers of BIS must be set up for this, or else the RDM would have to be introduced into an existing standardisation organisation.

Need for education

10300 No fundamental research is necessary for employing data structures for the objective of assessment, assuming all previously discussed features. Attention should, however, be given education and training in the area of data structures. Both as assessment instrument and as basis for the development of information systems, this structured approach proceeding from data has a large added value.

10301 Although in a specialist areas like computer science, a lot of attention is normally given to the data-oriented approach, the (complex) subject of data structures is usually avoided in the broader education programmes (like business management or economics).

10302 It is precisely learning to think systematically and in a structured way about information systems, in which primarily data structures are strong. Learning about data structures and how to design and use them creates skills such as understanding, logical reasoning, and insight in, in this case, information systems. These skills are somewhat comparable to the role of mathematics as means of teaching secondary school students some basic skills.

10303 *Who does not have anything to do with information systems these days?* Also independent of the process of software selection, it is increasingly important to quickly and correctly be able to use these systems. The information system is the backbone of most companies. More understanding of these systems could give important advantages during:-

(a) the introduction of systems, especially in the acceleration of the training;
(b) the development of systems, precisely where the future users have to help think about the specifications and requirements, and have to assess the design;
(c) the use of systems, in particular in the prevention of errors and the improvement of the efficiency of use.

10304 The (education) objective should be that one learns to understand how systems work instead of having to memorise tricks to carry out the routine actions.

Appendixes

A | Terminology in the area of data modelling

Data structure
A model of a set of data that describes the actual structure of this set. This (logical) structure is independent of both the way in which it is used by different applications, as well as the software and hardware techniques that are used to store, project, retrieve, access this set. The term data model is considered to be a synonym.

Entity
A 'thing' for which data is collected and recorded, such as objects, persons, abstract concepts or events, because it is relevant for the organisation.

Entity type
A set of entities that display the same behaviour and characteristics within a certain level of abstraction.

Relationship type
A logical association between entity types.

Attribute type
An important or relevant feature or characteristic of an entity type.

Generalisation / specialisation
The combination of a number of entity types into one single all-embracing type. Generalisation leads to a hierarchical structure of entity types. Specialisation is the opposite of generalisation.

Normalisation
The performance of a number of (theoretically and conceptually unambiguously defined) operations based on the dependence of attribute types, in order to come to an optimal description of entity types. Here 'optimal' means that no redundant attribute types or relationship types remain after normalisation.

Redundancy
The presence of elements (entity types, attribute types, or relationship types) that do not add any information and that are only present for either technical reasons or to increase the transparency.

Cardinality
A specification of the nature of a relationship type. In the RDM this cardinality is in principle limited to '1:n' and '1:1'. A further specification could have to do with the domain of n, for example whether or not 'n=0' is allowed.

B | Notation conventions for data structures

In the description of the Reference Data Model (RDM) a relatively simple notation technique is used. Figure B.1 shows a fictitious data structure in which a number of entity types is modelled to explain this technique.

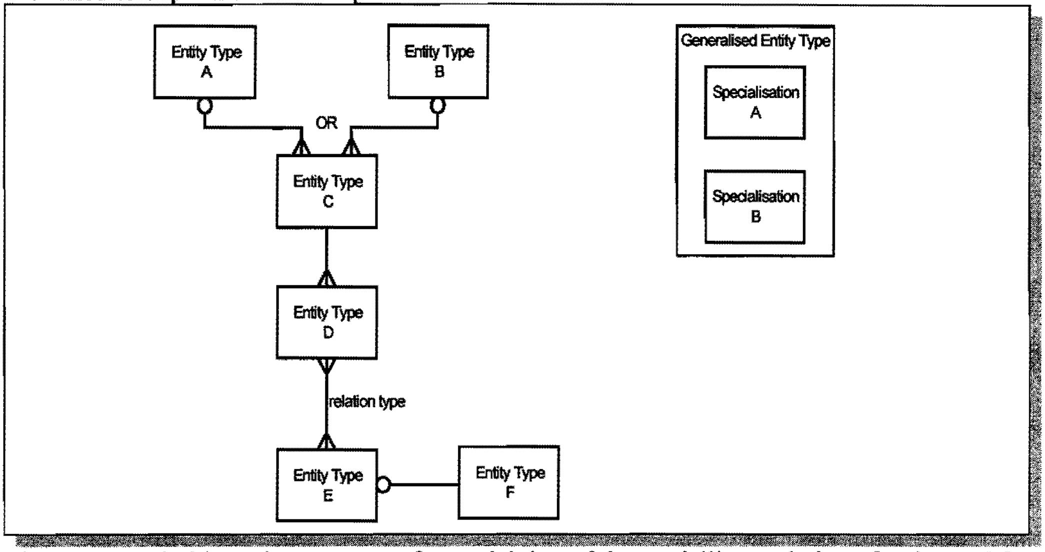


Figure B.1 - Fictitious data structure for explaining of the modelling technique for the RDM

Two *optional* 1:n relationship types are drawn between the entity types 'A' and 'C' and between the entity types 'B' and 'C'. This shows that the entity of the type 'C' may contain a reference (relationship) to the entity of type 'A'. The optional character of this relationship is indicated by the symbol 'O' in this figure.

A *mandatory* 1:n relationship type is drawn between entity types 'C' and 'D'. In contrast to the above, where the entity of type 'D' must be related to one entity of type 'C'.

An n:m relationship type is drawn between the entity types 'D' and 'E'. This indicates that an entity of type 'D' can be related to several entities of type 'E', and vice versa.

An *optional* 1:1 relationship type is drawn between the entity types 'E' and 'F'. This indicates that a single entity of type 'E' must be related to one entity of type 'F'. The converse relationship is optional.

The two optional 1:n relationship types at the top of this figure are exclusive, as indicated by an 'OR'. This shows that an entity of type 'C' is related either to an entity of type 'A' or to an entity of type 'B', but not both. These relationship types have to be optional, otherwise the 'OR' is not possible.

Relationship types can be named explicitly, as the relationship type between the entity types 'D' and 'E' in this figure. This shows the definition of the relationship type concerned. However, in many cases this definition is trivial and will therefore be skipped.

Finally, in the right part of the figure a specialisation of entity types towards two subtypes 'A' and 'B' is shown.

C | # Questionnaire

This appendix contains the complete questionnaire for the sections 'State-independent Manufacturing Data' and 'Purchasing', as published in [LOG94]. Because of the limited size of the questionnaire in the area of 'Purchasing', a number of supplementary questions has been added that are taken from the 1996 version [C&L96].

The questions are classified according to the 'data-process' typology from chapter 7. The types that are used can be summarised briefly as follows:

[1]**Type**	
Data-addressing	Process-addressing
DS = Data Singular	PP = Process Presence
DM = Data Multiple	PSP = Process Step Presence
DC = Data Complex	PSS = Process Step Sequence
	PSB = Process Step Behaviour

Further, every question is accompanied by a reference to the relevant figures in the RDM that treat the subject concerned, of course only where a relation exists with data structures.

A small number of questions has been found not or no longer, relevant. A brief argumentation for this is included in 'remarks'. These questions were not used in this research. Furthermore, they have either been removed or drastically reformulated in the new version of this questionnaire.

Where a question could be ambiguously interpreted, a short explanation is given in the 'remarks' regarding the interpretation of that question.

3	State-independent Manufacturing Data	Type[1]	Figure	Remarks
	Materials			
3.1.1	When adding new materials are the accompanying ledger accounts also validated?			This functionality is not very meaningful since general ledger accounts are usually defined at a higher (different) level of abstraction than materials. General ledger accounts (as stock accounts) are usually defined per item group. BIS often use a number of intermediate levels to 'calculate' the correct general ledger account, such as movement types and type of material. The next version of the publication (1996) no longer includes this question.
3.1.2	Is it possible to have multiple descriptions per item?			This question does not address the functionality sufficiently. Even if it means that one should be able to define different descriptions for various goals that are explicitly formulated (purchasing, sales) that is not to say that it is a discriminating criterion for selection. The next version of the publication (1996) no longer includes this question.
3.1.3	How many free data fields are available for item specification of classification codes?		6.28	This question has not been formulated specifically enough. What it should ask for are at least (combined) search and sorting facilities. If classification is important then a hierarchical coding and search structure is usually required. This formulation has been adapted in the next version of the publication (1996).
3.1.4	Is it possible to define alternatives for an item?	DM	6.18	
3.1.5	Is it possible to record different potencies or grades for one item per: I = item code; M = multiple items codes; L = item + lot number; R = item + release number?	 DS --- DS DM	6.47	This question combines some essentially different problem areas. Potency and grades are different material characteristics that require different solutions. Moreover, a 'solution' based on different items is not distinguishing. Any software package can do this but it is not a solution at all. Recording at item level is only meaningful if it concerns default values or standards (which is useful but as such insufficient to deal with these aspects).
3.1.6	Are non-stock items such as energy or steam stored in the item master file?			This is a question asking for the wrong solution or actually a work around. This one is always an option but does not always giving the required result. This formulation has been adapted in the next version of the publication (1996).
3.1.7	If one component is both a purchasing and a manufacturing item, can different lead/ delivery times be recorded?	DS	6.13	
3.1.8	Can an item have different units of measurement (pieces, weight, volume, area, length)?	DM	6.3	This question does not address the essence of the problem; any BIS can do this. What is important is the question whether deviating units can be used e.g., production registration in a different unit from the stock unit. Moreover, it is important to know whether the conversion factors between units that are not constant but can be different per item or even lot. This question has been interpreted as a question about the **availability of item-dependent conversion factors.**

3	State-independent Manufacturing Data	Type[1]	Figure	Remarks
3.1.9	Is an automatic procedure for conversion of these units of measurements available?	PSP		Mediocre question; see last remark.
3.1.10	Can multiple units be used per item like pieces, boxes, pallets?			Depending on the solution, these are defined in attributes of the item or there is a separate conversion table. Hence this question overlaps the interpretation of 3.1.9 and has therefore been skipped.
3.1.11	How is unit conversion recorded: I = in the item master file; B = in the bill of material; T = in a separate table; N = not at all?	 DS DS DM	6.3	(I) Units in the item master file can only be recognised as attributes. The application is rather limited anyway (e.g., only for a unit of packaging). (B) Basically, it is always possible to convert via bills of material. In that case, a (dummy) production operation has to take place as well. Mostly this will not be an acceptable solution. (T) In a normalised implementation (as in almost all BIS) this takes place via a separate table. What does matter is whether this table is or is not item- or even lot-dependent.
3.1.12	Apart from the item code, can the suppliers catalogue number also be used as a key: P = in purchasing; A = in all modules; or N = not at all?	PSB		The importance of this question is debatable. Whether or not it is, the availability of the mentioned attribute type will not reveal anything about the functionality required.
3.1.13	Apart from the item code, can the customer code also be used as a key: S = in sales; A = in all modules; or N = not at all?	PSB		See previous question.
	Bills of material			
3.2.1	How many bills of material (construction, manufacturing) are possible for one item?	DC	6.9 6.14	This question misses the essence of multiple bills of material. Different objectives apply to the use of different bills. These differ in such a way that they require separate questions and solutions (see the discussion in the RDM).
3.2.2	Are alternative items possible in the bills of material?	DM	6.18	
3.2.3	Are 'feed-ins' of items per operation possible without decoupling the operations by means of a semi-finished item?	DC	6.7	
3.2.4	Can a diverging structure with by-products be defined?	DC	6.40 6.41	

3	**State-independent Manufacturing Data**	**Type[1]**	**Figure**	**Remarks**
3.2.5	Can by-products have the same item number as the raw material (a solvent or catalyst is added in a loop)?	PSB	6.41 6.40	This will only work if the release of the by-product is properly timed. If *'by-product'* is a separate entity type one may expect this functionality to be available. The data structures do not directly show this. If real co-products have been implemented then this functionality can be expected to be almost certainly possible.
3.2.6	Can sets of spare parts be defined by means of a bill of material whereby delivery of the complete set is forced?	PSB		Although this functionality is related to the *'Sales Bills'* (see also paragraph 6226) such a bill of material not enough to support it. This functionality should be realised by means of a special attribute. One may expect that such an attribute type cannot be recognised very easily.
3.2.7	Can the material requirement be defined as: B = batch (fixed quantity); D = discrete (variable quantity)?	 DS DS		
3.2.8	Is scrap be stored in the bill of material as: B = batch (fixed); D = discrete (variable = additional percentage); or N = not al all?	 DS DS		
3.2.9	Can scrap be stored per operation as: B = batch (fixed); D = discrete (variable); or N = not at all?	 DS DS		
3.2.10	Can the material requirement be stored as a quantity (potency) of the active ingredient of the raw material?	DS		Usually, the requirement is expressed as 100%. Hence, it does not have to be stated in a separate bill of material relation. If necessary, one could mention that a different standard percentage can be defined per item. This functionality is considered to be available if the potency can be recorded per lot (of raw material).
3.2.11	Can the material requirement be defined in terms of quality (grade) of the active ingredient of the raw material?	DC	6.47	
3.2.12	Is it possible to record options on standard items in a variant bill of material?	DM	6.23	
3.2.13	Is it possible to record acceptable and unacceptable and inadmissable combinations for variant items?	DC	6.23	

3	State-independent Manufacturing Data	Type[1]	Figure	Remarks
3.2.14	Is a product configurator available to generate customer-related variant bills of material, based on the functional requirements of the customer?	PP	6.26	
3.2.15	Are unacceptatble options then automatically excluded?	DC	6.26	
3.2.16	Can order or serial number related configurations be stored in a history file?		6.22	This question falls outside the scope of the RDM. By the way, it does not formulate the essence clearly (see also the discussion on defining variant items). The next version of the publication (1996) no longer includes this question.
	Machines			
4.1.1	Can the normative capacity per machine be defined as: R = production rate (output per unit of time); or U = amount of time per unit of product?	DS		The distinction made in this question is not really relevant. A definition of any production speed per machine is sufficient in a situation in which the run time is *not* dependent on the operation (this is only an issue in cases of rather exceptional manufacturing processes).
4.1.2	Is the machine calendar defined as a week schedule per (group of) machine(s)?	DM	6.4	This question has been interpreted as defining a calendar per machine without distinguishing any possible forms.
4.1.3	Can one record in the machine calendar per machine: (1) a freely chosen (changing) shift per machine/day; (2) standstill because of preventive maintenance?			This question is not clear. Idle time as a result of maintenance means that the calendar has been adapted, but this can be done in every BIS.
4.1.4	Can the normative capacity be defined in freely chosen dimensions (number of people, m^2, m^3, kg, tonnes, kW)?	DS		This is merely a feature of *'Capacity Unit'* in the RDM.
4.1.5	Can the availability of this dimension (in quantity) freely be defined within the machine calendar?	DS		The question here is whether this attribute type is not defined in the *'Capacity Unit'* but in the calendar so that it can be time phased.
4.1.6	Are 'non-stock items' such as energy, steam etcetra be recorded as: I = item; M = machine; D = machine with a dimension with availability in the machine calendar?			A not very well formulated question. The question is whether two-dimensional capacity resources are supported (see previous question) or perhaps other functionality concerning 'non-stock' resources. Recording them as an item or machine is always possible, but does it deliver the functionality required?

3	State-independent Manufacturing Data	Type[1]	Figure	Remarks
4.1.7	Are storage tanks as a source of capacity with their dimension and availibility recorded in the machine calendar?	DC	6.37	Apart from the fact that this is a two-dimensional capacity resource, the content of this resource needs to be retained outside the scope of a production order. This means a link between *'Capacity Resource'* and *'Warehouse Location'*. Nevertheless is no question as to this functionality. However, the question has been interpreted as such during the assessment.
4.1.8	Are internal means of transportation (fork lift trucks) with their availability recorded in the machine calendar?			The essence of transport resources is not expressed by this question. The way it is formulated here treating it as a machine is sufficient, and this functionality already becomes clear in the previous questions. The next version of the publication (1996) no longer includes this question.
	Equipment and tooling			
4.2.1	Are tools recorded in: I = item master file; M = machine file; T = separate tooling file?	 DS DS DM	6.36	This question was formulated more or less as a check. Considering tools as items or machines is always possible but often results in limited functionality. The essence of the question is whether tools and their special functionality are being supported. In order to try and cover this up the following questions were asked as to special attribute types and relations with other entity types.
4.2.2	Are the following data recorded for tools: D = delivery time; M = in/out maintenance; O = operating time; R = remaining operating time?	DS		
4.2.3	Are tools with a relational record (m:n) directly linked with: O = operation; M = machine; P = product; C = customer; or N = not al all?	 DC DC DC DC	6.36	
4.2.4	What is the number of tools that can be assigned to an operation: 1; L = list of tools; B = bill of tools; or N = not at all?	 DS DM DC	6.36	

3	**State-independent Manufacturing Data**	**Type[1]**	**Figure**	**Remarks**
4.2.5	Can alternative tools also be assigned as 1; L; B; or N?	DS DM DC	6.36	
4.2.6	Are sequence dependent adjustment times recorded in: C = codes per group; T = input/output tables; or N = not at all?	DM	6.46	It is not clear what the codes per group are. The availability of sequence-dependent adjustment times seems to be discriminating enough.
4.2.7	Can different operations be added for forging and spray casting in the bill of operations of a product: T = tool preparation; with a separate M = materials requirement?			This question is not clear and is no longer included the next version of the publication (1996).
	Operations			
4.3.1	Is it possible for recipes to enter operations, bill of material relations and tools on one data entry screen?			When we consider the present technology, including Windows-oriented applications, these types of question are of no relevance any longer.
4.3.2	Can free text blocks be added to recipes?	DS		The importance of this question is debatable in view of modern (database) technology.
4.3.3	Are operations automatically adjusted on the basis of: P = Potency (concentration of active matter); or G = Grade (quality of the raw material)?	 PSB PSB		When defining the requirements of material the potency or grade might also be included, bus this does not justify the conclusion that run times are adjusted accordingly. It is impossible to draw any conclusions on this on the basis of data structures.
4.3.4	Can standard routings (bills of operations) be recored that can be used for several products?	DC	6.16 6.17	
4.3.5	Can the usage of a resource be defined in the matching dimension (amount of people, m^2, m^3, kg, tonnes, kW)	DS		This is about defining the requirement of two-dimensional capacity resources, in other words: besides the required run time (the first dimension) also a second requirement dimension (e.g.. m^2 space).
4.3.6	Can succeeding operations on a by-product be defined without decoupling via a semi-finished product or intermediate stock?	DC	6.41	

3	**State-independent Manufacturing Data**	**Type[1]**	**Figure**	**Remarks**
4.3.7	Can parallel operations be defined?	DS		What is meant here is an overlap in operations through an indication stored in some attribute and not via a normalised network structure (see also parapraph 6327). This also becomes clear from the following questions.
4.3.8	In that case, can an overlap percentage be assigned as: T = time; A = amount of products; or N = not at all?	DS DS		
4.3.9	Are 'Feed-ins' of operations possible so that a convergent network of operations is created without defining operations as parallel?	DC	6.45	
4.3.10	Is a divergent network of operations possible without parallel operations or decouplings via semi-finished products or stocks?	DC	6.45	
4.3.11	Can the following relations be defined in the operations record: T = time; A = amount of products; or N = not at all? (1) End-Start $_{min}$ (cure time) (2) End-Start $_{max}$ (pot life) (3) Start-Start $_{min}$ (minimum overlap) (4) Start-Start $_{max}$ (maximum overlap)	DS DS DS DS		This becomes clear from the availability of these attribute types, although the availability of a real network structure is already an important indication.
4.3.12	Can a 'fixed' pumping time be defined between a reactor and a tank?	DS		A fixed time can be realised by equalling a minimum and maximum (see also the previous question).
4.3.13	Are alternatives with different operation, or lead times possible for: (1) different machines (same operation)? (2) different operations? (3) subcontracting?	DC DC DC	6.20 6.32	
	Engineering changes			
4.4.1	Is a standard procedure available for changes in bills of material, bills of operations etcetera, such as: M = mass replace; R = restricted replace?	PSP		This is pure functionality. As a result of a selection the selected bills are adapted simultaneously.

3	**State-independent Manufacturing Data**	**Type[1]**	**Figure**	**Remarks**
4.4.2	Can changes (mass replace) be made to bills of material and bills of operations in terms of: effective date; customer number; serial number; 'if the old material is finished'; 'as soon as the new material is available'?	 DC DC DC PSB PSB	6.22	This question is interpreted as being able to deal with several versions of 'bills' which may temporarily exist next to each other, in conformity with the relation or criterion mentioned in this question.
4.4.3	Are the issue numbers of bills of material automatically increased when changes are made?	PSB		The usefulness of this functionality is not really clear.
4.4.4	Is the referential integrity of the data base safeguarded when changes are made?			This question addresses the database management system and falls outside the scope. The question is rather about the incorrect functioning of the system. The next version of the publication (1996) no longer includes this question.

6	Purchasing	Type[1]	Figure	Remarks
6.1.1	Is an integrated purchasing procedure available?	PP		This question is rather trivial but this seems to be intrinsic to all questions of the 'PP' type.
6.1.2	Can purchasing take place on (customer) order (item not recorded in the item master file)?	DC	5.5	In this case, a wrong combination is being made. On the one hand, the question is about purchasing on customer order but these can be standard items. On the other hand, the question is about one-off customer-specific items that are bought. We have interpreted the question as one about the purchasing of customer-specific items which are not individually recorded in an item master file.
6.1.3	Is it possible per purchase order line to state: T = a separate delivery time; S = a separate delivery scheme?	 DS DM	5.16	
6.1.4	Is it possible per purchase order line to record: R = required; P = promised; R = realised delivery time?	 DS DS DS		Actually, the delivery time realised has nothing to do with the purchase order line. It is an element of good receipts. Moreover, it is a rather trivial feature. The question has been interpreted as a highly meaningful one as to whether there is a distinction between the required and the agreed delivery time.
6.1.5	Are blanket orders and call-offs possible?	DM	5.15	
6.1.6	Can these delivery schedules be changed afterwards?			The purpose of this question is not clear; such functionality is trivial. The next version of the publication (1996) no longer includes this question.
6.1.7	Is a procedure for return deliveries available?	PP	5.17	
6.1.8	Is a procedure for subcontracting available?	PP	5.5	
6.1.9	Must subcontracting take place on the basis of: M = material code; or O = operation code?	DC	5.5	In fact, subcontracting on the basis of a material code concerns the subcontracting of the total manufacturing of the required material, while subcontracting on the basis of operation code concerns the subcontracting of individual operations. This question is not discriminating as to the first form of subcontracting and can be regarded as 'normal' purchasing functionality.
6.1.10	Are two different delivery times available for the options: own manufacturing and subcontracting?	DS	5.5	
6.1.11	Is a procedure available for material delivery to subcontracting firms?	PSP	5.6	This question is not formulated correctly. As a matter of fact the delivery of material to the subcontractor is an important feature. The question formulated the way it is, will always be possible. This is a typical example in which the implementation chosen is or is not acceptable, depending the specific requirements. We have interpreted this question as whether it is possible to record the material delivery of operations to be subcontracted (purchased).

6	Purchasing	Type[1]	Figure	Remarks
6.1.12	Is expediting (before delivery) possible?			By means of a report generator such functionality is always possible. Some software packages have special functions but this is not made clear by the question. The next version of the publication (1996) no longer includes this question.
6.1.13	Is a dunning procedure available?			This question addresses dunning in arrear, in case the goods should have been received. See our remark on the previous question. The next version of the publication (1996) no longer includes this question.
6.1.14	Is monitoring the supplier quality (vendor rating) possible for: T = delivery time; Q = quality; D = delivery quantity; No = none?	DM DS DS DS	5.22	
6.1.15	Is performance monitoring of buyers possible in terms of: T = delivery time; Q = quantity; P = price; No = none?	DM DS DS DS	5.22	
6.1.16	Is a procedure in place for inspection of incoming goods?	PSP	5.17	
6.1.17	Is discount monitoring possible?			It is not clear which functionality is meant. The next version of the publication (1996) no longer includes this question.
6.1.18	Is it possible to accumulate discounts, for example, quantity, shipment and annual bonus?	PSP		This question has been interpreted as one about the possibility to (separately) financially process discounts and bonuses.
6.1.19	Is there a standard provision for the electronic receipt and transmission of purchase orders / call-offs, order confirmations, invoices?	PSB		
6.1.1 (1996)	When purchasing on sales order is the sales order number recorded in the purchase order?	DC	5.18 5.29	In addition to question 6.1.2 this question inquires after the relation between customer orders and purchasing orders.

6	**Purchasing**	**Type**[1]	**Figure**	**Remarks**
6.1.3 (1996)	Are project purchases possible in order to automatically book the costs on a project when receiving goods?	DC	5.18	
6.1.15 (1996)	Are various ship-from addresses per supplier possible?	DM	5.3	
6.1.16 (1996)	Can one distinguish between the shipping dat from the supplier and the date of receipt in the warehouse?	DS		
6.1.17 (1996)	Can quotations from suppliers be recorded?	DM	5.10	
6.1.18 (1996)	Can a request for quotattion be recorded to be sent to multiple suppliers?	DC	5.10	
6.1.19 (1996)	In a multi-site environment can one record which site purchases a certain item?	DC		
6.1.20 (1996)	Can a fixed carrier be recorded per: I = Item; T = item type; S = supplier; or N = not at all?	 DS DS DS		
6.1.21 (1996)	Are multiple (alternative) item-supplier combinations possible?	DC	5.23	

Samenvatting (summary Dutch version)

1 Er is een duidelijke trend waarneembaar verschuivend van het zelf ontwikkelen van bedrijfsinformatiesystemen (BIS) naar het aanschaffen van standaardpakketten. In steeds meer branches vormen standaardpakketten een goed alternatief. Deze ontwikkeling is met name in gang gezet rond de systemen voor productiebesturing (de MRP-pakketten), die zich enerzijds ontwikkelen naar systemen voor besturing van de gehele logistieke keten en anderzijds naar ondersteuning van *alle* bedrijfsfuncties, dus bijvoorbeeld ook financiën en personeelsmanagement.

2 De huidige trend naar standaard-BIS wordt wel beïnvloed door de snelle ontwikkeling van de technologie. Zo zijn grafische user interfaces uitgegroeid tot een belangrijk selectiecriterium. Object oriëntatie is in het bijzonder een technologie die het in de toekomst misschien mogelijk maakt om losse objectklassen van verschillende leveranciers te combineren tot een integraal BIS, bijvoorbeeld een inkoopobjectklasse van de ene leverancier met een voorraadobjectklasse van een andere leverancier.

De juiste keuze van een standaard-bedrijfsinformatiesysteem (BIS)

3 Voor een onderneming is het BIS een kritieke succesfactor, succes of falen van de invoering ervan beïnvloedt de concurrentiepositie en het rendement. Eerst moet echter een keuze worden gemaakt uit een breed aanbod van kandidaat-BIS. Gezien de diversiteit van ondernemingen en hun karakteristieken moge het duidelijk zijn dat een juiste keuze essentieel is.

4 Voor het selecteren van complexe systemen is op basis van ervaringen een algemeen geaccepteerde aanpak gegroeid. Deze 'best practice' aanpak bestaat uit een aantal fasen en activiteiten, met als belangrijke stadia:-

(a) *de voorselectie*, waarbij op basis van een zeer beperkte inspanning een longlist van minder dan een tiental kandidaten wordt opgesteld;

(b) *de detailselectie*, waarbij deze longlist wordt ingekort tot een shortlist van niet meer dan een drietal BIS. Deze selectie vindt plaats op basis van een beperkt aantal essentiële beoordelingscriteria, ook wel knock-out-criteria genoemd;

(c) *de toetsing*, waarbij door middel van een bedrijfsspecifieke case de shortlist-kandidaten in een demonstratie concreet worden getoetst;

(d) *de afronding*, doorgaans gericht op het afsluiten van het contract en het voorbereiden van de implementatie.

5 Niet het enige, maar wel een overheersend element bij de selectie is het beoordelen en vergelijken van de functionele kenmerken van de kandidaten. In de geschetste aanpak gebeurt dit op basis van (gebruikers-)eisen en wensen die getoetst worden in een vragenlijst en casedemonstraties.

Zowel het opstellen als het beoordelen van deze vragenlijst is geen sinecure. De vragen blijken veelal op meerdere (soms creatieve) manieren geïnterpreteerd te kunnen worden. Ook wordt er vaak gevraagd naar een specifieke oplossing in het BIS terwijl alternatieven - waar men misschien niet aan gedacht heeft - ook mogelijk (of zelfs beter) zijn. Aan de andere kant, als gevraagd wordt of de leverancier een oplossing biedt voor een geformuleerde probleemstelling, dan zegt een bevestiging van de leverancier nog weinig over de kwaliteit.

Bijvoorbeeld: „*Kan materiaal op klantorder worden ingekocht?*"; een „*Ja*" van de leverancier zegt echter nog niets over de wijze waarop dit kan, het aantal handelingen en de eenvoud, en onder welke functionele voorwaarden.

6 Ook de toetsing door middel van de bedrijfscase is zeker niet waterdicht. Zo is het opstellen van een goede testcase met alle bijzondere bedrijfs- en proceskarakteristieken niet eenvoudig. Daarna is het in één of twee dagen demonstreren van zo'n complexe case aan de aanstaande gebruikers wel mogelijk, maar leidt dit niet tot een objectief oordeel van deze gebruikers. In plaats daarvan blijken belangrijker te zijn: de commerciële vaardigheden van de demonstrateur, het uiterlijk van het systeem en de mate waarin het systeem aansluit bij het bestaande (oude) referentiekader van de beoordelaar.

7 Zeker gezien de complexiteit en breedte van de BIS waar het hier om gaat, zijn vaak vele weken (of zelfs nog meer) nodig om dergelijke systemen goed te beoordelen. In feite wordt dit veroorzaakt door de 'black-box'-achtige benadering van het op deze wijze beoordelen van systemen. Deze benadering maakt namelijk geen gebruik van enige kennis van de interne structuur van systemen.

De doelstelling van het onderzoek

8 Dit onderzoek is ontstaan vanuit de indruk dat het beoordelen van de datastructuur van BIS een belangrijke toegevoegde waarde heeft voor de kwaliteit van selecties. Het bestuderen van de datastructuur als kern van een BIS, is te beschouwen als het keuren van de techniek onder de motorkap van een auto. *Het onderzoek is dan ook gericht op de mogelijkheden, maar ook de beperkingen van het gebruik van datastructuren bij het beoordelen van bedrijfsinformatiesystemen (BIS).*

De methode voor het beoordelen van datastructuren

9 Ten behoeve van het onderzoek is een methode ontwikkeld om de datastructuren van verschillende BIS op een geüniformeerde wijze te vergelijken. De kern van deze methode is een referentiedatamodel (RDM). In plaats van het onderling vergelijken van de zeer verschillende datastructuren van diverse BIS, worden in de RDM-methode deze datastructuren eerst geprojecteerd op dit RDM. Op basis van deze projecties worden conclusies getrokken over de aanwezigheid van functionaliteit in het betreffende BIS, alsmede de aard van de geboden oplossing en implementatie. Daarnaast kunnen de projecties direct met elkaar worden vergeleken, zodat de beoordelaar zich slechts hoeft te richten op de verschillen.

10 Voor een aantal functiegebieden is een bruikbaar RDM ontwikkeld, namelijk voor '*inkoop*' en voor '*toestandsonafhankelijke productiegegevens*' (zoals stuklijsten, routings en recepturen). De uitwerking van dit RDM beslaat een substantieel deel van het onderzoek. Ter illustratie worden op dit RDM diverse (deel-)projecties van bestaande BIS getoond. Het RDM is voor een groot deel ontwikkeld door middel van het combineren van een groot aantal datastructuren van bestaande BIS waaronder Prism, Ratio, SAP-R/3, Triton en TXBase. De resultante is daarmee ook de bundeling van kennis en ervaring op de betreffende functionele gebieden zoals die heden ten dage door verschillende onafhankelijke leveranciers is opgebouwd.

11 Het RDM kent een aantal specifieke doelstellingen die het wezenlijk doen verschillen van een datastructuur van een te ontwikkelen informatiesysteem, namelijk:-

(a) *het definitievermogen*, dat een eenduidig begrippenkader moet geven;

(b) *het projectievermogen,* waardoor totaal verschillende datastructuren toch projecteerbaar zijn. Dit betekent *bijvoorbeeld dat het RDM alternatieve oplossingen bevat die normaliter niet in één logische datastructuur worden gecombineerd;*

(c) *het onderscheidende vermogen,* zodat bij verschillende datastructuren ook na projectie de belangrijke verschillen zichtbaar blijven. Dit betekent in veel gevallen dat de datastructuren in het RDM vrij gedetailleerd (of gespecialiseerd) zijn;

(d) *het inzichtelijke vermogen,* waardoor de filosofie van het oorspronkelijke BIS zichtbaar is.

12 Een principiële constatering van dit onderzoek is dat het op deze wijze ontwikkelde RDM geen datastructuur is - of kan zijn - van een nieuw (allesomvattend) BIS en dus ook beslist niet mag worden verward met een universeel datamodel (UDM). Kenmerkende verschillen tussen het RDM en een UDM zijn:-

(a) het UDM kiest uit alternatieven, waar het RDM alternatieven combineert;

> *Door het combineren van datastructuren zal het RDM automatisch meerdere alternatieven voor eenzelfde functie omvatten, bijvoorbeeld zowel een productiemodel als de traditionele routings en stuklijsten. In een UDM dient de principiële keuze gemaakt te zijn voor één alternatief voor de betreffende doelgroep.*

(b) het UDM vermijdt redundantie, waar het RDM deze juist benut;

> *Redundante gegevens (of relaties) zijn in een BIS bij uitzondering gewenst. Zo wordt de ene keer eerst een bewerking vastgelegd waaraan de materiaalbehoefte wordt gekoppeld, terwijl de andere keer wordt uitgegaan van de materiaal-stuklijst die pas later wordt gerelateerd aan de bewerkingsstructuur. Deze combinatie van mogelijkheden leidt tot een aantal redundante relaties tussen gegevensgroepen. In het RDM treedt redundantie structureel op als gevolg van het volledig combineren van datastructuren, ook waar dit in een concreet informatiesysteem geen toegevoegde waarde heeft. Een informatiesysteem met het RDM als datastructuur zou leiden tot onnodig veel extra handelingen en invoer van gegevens.*

(c) het UDM is logisch consistent, waar het RDM dit niet hoeft te zijn en vanwege de gevolgen van (a) en (b) waarschijnlijk ook nooit is.

13 Vanwege het combineren van datastructuren van BIS met een zeer brede reikwijdte, beperkt het RDM zich niet tot één branche, logistieke besturingsvorm of type productieproces. Dit maakt het RDM waardevol voor het vergelijken van verschillende situaties of typologieën, maar levert bijvoorbeeld geen branchemodel op.

De toetsing van het beoordelingsinstrument 'datastructuren'

14 In het kader van dit onderzoek is een toetsing uitgevoerd naar de kwaliteit van het oordeel op basis van datastructuren. Hiertoe is deze beoordelingswijze vergeleken met de antwoorden van leveranciers op een traditionele vragenlijst.

15 Er is gebruikgemaakt van een bestaande vragenlijst die was opgesteld voor het inventariseren van softwarepakketten voor productiebesturing. Deze publicatie, die grote bekendheid geniet op de Nederlandse markt, bevat van alle leveranciers de antwoorden op een uitgebreide lijst van vragen naar de functionaliteit van de door hen geleverde BIS. Bij het inventariseren werden de leveranciersantwoorden gescreend door onafhankelijke adviseurs.

16 Voor een viertal BIS is op basis van de datastructuur eveneens een oordeel gevormd over deze vragen. **In een substantieel aantal gevallen blijkt dit oordeel af te wijken** van de leveranciersbeantwoording. Ter verificatie zijn deze verschillen met de betreffende leveranciers besproken. Van de meeste verschillen heeft de leverancier bevestigd dat de conclusie op basis van de datastructuur juist was, hetgeen overigens niet wil zeggen dat in al deze gevallen de betreffende antwoorden op de vragenlijst fout zouden zijn. De belangrijkste oorzaken van deze verschillen bleken:-

(a) een interpretatie door de leverancier die afwijkt van de eigenlijke intentie van de vraag (met enige creativiteit blijken veel vragen voor meerdere uitleg vatbaar);

(b) bijzondere 'work-arounds' (of trucs) die toevallig in de software mogelijk zijn, waardoor enige functionaliteit gerealiseerd kan worden die lijkt op het gevraagde. In hoeverre dit inderdaad voldoende is zal van situatie tot situatie verschillen;

(c) versieverschillen. Dit betekent dat zowel het antwoord van de leverancier als de conclusie op basis van de datastructuur op zich correct zijn. Door het opmerken van dergelijke verschillen wordt het beoordelingsvermogen van datastructuren wel bevestigd;

(d) onvolkomenheden in de documentatie van enkele van de beoordeelde BIS.

17 Teneinde de conclusies te relateren aan het type vragen, zijn de vragen geclassificeerd. De hoofdindeling van deze classificatie onderscheidt gegevensgerichte en procesgerichte vragen. De gegevensgerichte vragen zijn direct vertaalbaar naar enkelvoudige, meervoudige of complexe datastructuren. De procesgerichte vragen zijn gericht op de ondersteuning van een proces of van een activiteit in een proces, op de ordening van activiteiten in een proces of op het gedrag van deze activiteiten.

18 Nadat de betreffende vragenlijst op basis van deze classificatie is ingedeeld, blijkt het percentage gegevensgerichte versus procesgerichte vragen respectievelijk 83% en 17%. Hieruit zou afgeleid kunnen worden dat vragenlijsten die door experts zijn opgesteld zich in het bijzonder richten op de datastructuur, overigens zonder dat dit zoals in dit onderzoek expliciet is gemaakt. In hoeverre deze veronderstelling voor elke vragenlijst zal gelden, waarbij tevens een uitspraak gedaan moet worden over goede en slechte vragen, is niet onderzocht.

19 Enkele kwantitatieve resultaten van de uitgevoerde toetsing:-

(a) het aantal geconstateerde verschillen waar de datastructuur tot een correcte conclusie kwam, is bijna 14% van de gegevensgerichte vragen;

(b) het percentage vragen waar op basis van de datastructuren geen uitspraak kon worden gedaan, is ongeveer 17%;

(c) het percentage gegevensgerichte vragen waar op basis van de datastructuren geen uitspraak kon worden gedaan, is zo'n 7%;

(d) in een tweetal gevallen werd door de datastructuur meer functionaliteit gesuggereerd dan in werkelijkheid geïmplementeerd bleek te zijn.

20 Bij de analyse van de vragen waarover datastructuren geen uitspraak doen, zijn twee categorieën te onderscheiden.

21 Ten eerste, de functionaliteit die vertaalbaar is naar enkelvoudige eigenschappen van de datastructuur, namelijk naar individuele attribuuttypen, kan vaak (maar niet altijd!) aannemelijk worden gemaakt aan de hand van de gedetailleerde data dictionary. Aan de *aanwezigheid* van attribuuttypen mag echter zeker geen definitieve conclusie worden verbonden ten aanzien van deze gewenste functionaliteit, het is slechts een indicatie. De *afwezigheid* is wel een stevige basis voor discussie met de leverancier hoe de gevraagde functionaliteit dan gerealiseerd zou zijn.

22 Ten tweede, voor een tweetal typen procesgerichte vragen is het meestal onmogelijk - overigens niet onverwacht - om op basis van de datastructuur uitspraken te doen. Het betreft met name vragen naar het *gedrag van individuele processtappen* en de *ordening van stappen in processen.*

23 De kwantitatieve resultaten (zie alinea 20) zijn afhankelijk van de specifieke vragenlijst waarmee de toetsing wordt uitgevoerd. Déze vragenlijst en de geverifieerde antwoorden geven een duidelijke indicatie van de sterke en zwakke kant van het gebruik van datastructuren. Dit wordt expliciet bevestigd door het relatief hoge percentage geconstateerde verschillen. Stel dat de toetsing zou hebben plaatsgevonden op basis van een 'perfect beantwoorde vragenlijst', dan zou de datastructuur niet tot een afwijkende constatering komen. Dit betekent echter alleen dat met deze vorm van toetsen het nut van datastructuren niet kan worden bevestigd. Er wordt immers vergeleken met een theoretische ideaalsituatie.

De evaluatie en conclusies

24 Datastructuren zijn een noodzakelijke voorwaarde maar geen voldoende voorwaarde voor de aanwezigheid van functionaliteit. Dit impliceert dat de alleen *afwezigheid* van de datastructuur tot een definitieve conclusie leidt.

25 Voor het aantonen van de *aanwezigheid* van functionaliteit geldt dat het beoordelen van de datastructuur alleen indirect iets zegt over de **mogelijk** aanwezige functionaliteit. Alleen een datastructuur zonder programmatuur levert niet de gewenste functionaliteit. Echter mede ondersteund door de ervaringen bij de toetsing, blijkt dit geen onoverkomelijk probleem voor de beoordeling van standaard-BIS. Als de datastructuur in de vorm van (complexe) entiteittypen en relatietypen aanwezig is, blijkt doorgaans ook de functionaliteit aanwezig te zijn. Het ontwerpen van zo'n complexe datastructuur is immers één van de belangrijkste ontwikkelactiviteiten. Gegeven deze inspanning is het bij de ontwikkeling van een BIS een gemiste kans om de overige ontwikkelingsactiviteiten niet uit te voeren. Aan de andere kant, mocht alleen dit fundament in de vorm van de datastructuur aanwezig zijn, dan is het bijbouwen van de gewenste functionaliteit relatief goedkoop.

26 Deze aanname ten aanzien van de aanwezigheid van functionaliteit geldt echter niet voor (eenvoudige) individuele attribuuttypen. Hier moet men voorzichtiger omgaan met het trekken van conclusies. Regelmatig worden attribuuttypen aangetroffen met als omschrijving 'voor toekomstig gebruik'.

27 De toetsing bevestigt dat datastructuren inzicht verschaffen in de functionele mogelijkheden van BIS. Dit inzicht wordt gedemonstreerd door de volgende bijzondere kenmerken ten opzichte van bijvoorbeeld een functionele vragenlijst:-

(a) de datastructuur geeft de context weer waarin de functionaliteit wordt ondersteund, dus met name welke combinaties van functionaliteit mogelijk zijn, dan wel elkaar juist uitsluiten;

(b) de datastructuur maakt de filosofie van het BIS zichtbaar, te weten de gebruikte terminologie, de aggregatieniveaus van entiteittypen, de gemaakte keuzen en de ondersteunde (productie-, inkoop- of bedrijfs-) structuur;

(c) een datastructuur kent meerdere abstractie- en beschrijvingsniveaus waardoor het mogelijk is om naar eigen behoefte in en uit te zoomen in deze beschrijvingsvorm;

(d) de datastructuur omvat veel meer informatie dan vooraf bedacht of geformuleerd zou kunnen worden. Waar de informatie uit een beantwoorde vragenlijst of een demonstratie beperkt is tot hetgeen vooraf of op dat moment relevant leek, kan uit de datastructuur ook achteraf nieuwe relevante informatie worden gehaald. Immers een datastructuur is een complete beschrijving van het gegevensaspect.

28 De belangrijkste beperkingen van het gebruik van datastructuren en in het bijzonder de RDM-methode zijn:-

(a) de expertise-eisen die gesteld worden aan de beoordelaar;

(b) de investering die nodig is om een goed RDM te ontwikkelen en de datastructuren van alle belangrijke BIS hier initieel op te projecteren (deze beperking geldt dus specifiek voor de RDM-methode en niet voor het gebruik van datastructuren in z'n algemeenheid);

(c) de onvolledigheid van de beoordeling. Er is een beperkte maar wel bekende categorie van kenmerken waarover geen uitspraak wordt gedaan.

29 Verder dient men zich te realiseren dat datastructuren oplossingen (keuzen en implementaties) beschrijven en niet bedoeld zijn voor het beschrijven van een probleemstelling (welke functionaliteit is gewenst). Voor één probleemstelling kunnen meerdere alternatieve datastructuren een oplossing leveren. Met andere woorden datastructuren zijn dus niet geschikt voor het formuleren van een algemene behoefte of gewenste structuur van standaard-BIS. De 'vraagstelling' is dus geen datastructuur, maar de datastructuur is het 'antwoord' op een vraag naar functionaliteit.

30 In de detailselectie, bij het opstellen van een shortlist van BIS, kunnen datastructuren een vergelijkbare rol vervullen als die van een database van BIS met functionele kenmerken. Eén van de voordelen ten opzichte van zo'n database is dat veel meer informatie beschikbaar is in datastructuren dan vooraf gedefinieerd kon worden. Bij de toetsing van de shortlist van kandidaat-BIS levert de datastructuur van deze BIS in de voorbereiding informatie over de kritieke punten die expliciet getoetst moeten worden in de demonstratie.

31 Op alle gebieden is de benodigde expertise voor het op de juiste wijze hanteren en interpreteren van datastructuren van cruciaal belang.

De toegevoegde waarde bij de selectie van BIS

32 Het op een juiste wijze gebruiken van datastructuren als beoordelingsinstrument, kan in het selectieproces van standaard-BIS de volgende toegevoegde waarde hebben:-

(a) *het systematisch herkennen van de kritieke punten* in (de datastructuur van) standaard-BIS;

(b) *het voortijdig onderkennen van de grote noodzakelijke aanpassingen (en daarmee het inschatten van de benodigde inspanning)*, die immers volgen uit de toetsing van de kritieke punten in de datastructuur;

(c) *het minimaliseren van de kans dat **onverwachte** fundamentele aanpassingen nodig zijn*, immers de datastructuur is geverifieerd en het fundamentele karakter van aanpassingen betreft meestal het wijzigen van de (data)structuur;

33 *een garantie op een passende (=juiste) keuze*, in de vorm van een passende datastructuur en daarmee basisfunctionaliteit.

References

[ANS75] ANSI/X3/SPARC; *Study Group on Data Base Management Systems: Interim Report 75-02-08*; ACM SIGMOND Newsletter, volume 7 no 2; 1975

[API92] APICS; *APICS Dictionary;* 7th edition; 1992

[BAC65] C.W. Bachmann; *Software for random access processing*; Datamation; 1965

[BAN94] S.K.Banderjee, A.P. Kourouklis, J. Penman; *Manufacturing planning and control decision model: A design methodology*; International journal of production economics; 1994

[BER90] J.W.M. Bertrand, J.C. Wortmann, J. Wijngaard; *Production control: a structural and design approach*; Elsevier Science Publishers; 1990

[BER92] A. Berson; *Client/Server architecture*; McGraw-Hill; 1992

[BOT80] C.H. Botter; *Industrie en organisatie* (Industry and organisation, only in published in Dutch); Kluwer; 1980

[BRU92] T.A. Bruce; *Designing quality databases: practical information management & IDEF1X*; IDEF1X1992

[BUT92] Butler Associates Ltd; *Implementor*; Dublin; 1992

[BUY92/95] BuySmart!™; *Expert Buying Systems*; 1992 / 1995

[CHE76] P.P.-S. Chen; *The entity-relationship model toward a unified view of data*; ACM Transactions on database systems; 1976

[COD70] E. Codd; *A relational model for large shared data banks*; Communications of the ACM; June 1970

[COM79] I. St. J. Hugo; *Guide to production control systems 1979/80*; Computer Guides Ltd; 1979

[COM81] B. Knight; *Guide to production control systems 1981/82*; Computer Guides Ltd; 1981

[CAU94] R.M.A. Caubo; *Pakketbeoordeling op basis van gegevensmodellen: De beschrijving van een referentiegegevensmodel voor stuklijsten en routing* (Software assessment based on data models: A description of a reference data model for bills-of-material and routings, only published in Dutch); TUE; 24 March 1994

[C&L80/85] Coopers & Lybrand International; *Manufacturing Resource Management (MRM): Manufacturing Systems Methodology*; unpublished; 1980 / 1985

[C&L87] E.S. Andersen, K.V. Grude, T.Haug, J.R. Turner; *Goal-directed project management*; Coopers & Lybrand; 1987

[C&L91] F.H. Huele, P.C.M. Mater; *Kwaliteit in logistiek beleid* (Quality in logistic policy, only published in Dutch); Kluwer Deventer; 1991

[C&L94] Coopers & Lybrand MCS; *Knock-out criteria voor het selecteren van pakketten voor productiebesturing* (Knock-Out Criteria for selecting manufacturing software, only in Dutch); unpublished; 1994

[C&L95] Coopers & Lybrand MCS; *Overzicht integrale logistieke en productie-besturingssoftware* (Survey on integrated software for logistics and production control, only in Dutch); unpublished; 1995

[C&L96] Logiplan and Coopers & Lybrand; *Standaardsoftware voor productiebesturing* (Standard software for production control, only published in Dutch); 1996

[DBTG71] DBTG; *Report of the CODASYL Data Base Task Group*; ACM; April 1971

[DEM79] T. DeMarco; *Structured Analysis and System Specification*; Prentice-Hall, Englewood Cliffs; 1979

[DIJK84] E.W. Dijkstra and W.H.J. Feijen; *Een methode van programmeren* (A method of programming, only published in Dutch); Academic service; 1984

[ELM89] R. Elmasri and S.B. Navathe; *Fundamentals of database systems*; Addison Wesley; 1989

[ESP91] ESPRIT Consortium AMICE; *CIM-OSA AD 1.0 Architecture description*; Brussel; 1991

[GAN83] H. Gand and M.E. Cook; *Choosing an MRP-system*; Datamation; January 1983

[GAR94.1] E. Keller; *ERP Vendor Guide*; CIM Strategic Analysis Report by Gartner Group; May 1994

[GAR94.2] E. Keller; *ERP Selection Process: Methodology and Vendor Comparison*; CIM Strategic Analysis Report by Gartner Group; June 1994

[GAR94/95] E. Keller; *CIM Research Notes*; Gartner Group; 1994/1995

[GAV94] C.J. Gavin and D. Little; *Application of CASE within manufacturing industry*; Software engineering journal; July 1994

[GRA91] P.A. Gray; *Open Systems: A Business strategy for the 1990s*; McGraw-Hill Book Company; London 1991

[HAM80] M.M. Hammer and D.J. McLeod; *Database description with SDM: A semantic data model*; ACM Transactions on database systems; September 1980

[HAR94] A. Hars; *Referenzdatenmodelle: Grundlagen effizienter Datenmodellierung*; Gabler wirtschaft; 1994

[HEE94] K.M. van Hee: *Information systems engineering: a formal approach*; Cambridge University; 1994

[HEG95] H.M.H. Hegge; *Intelligent product family descriptions for business applications*; Dissertation TUE; 1995

[HEI94.1] J.C.J. de Heij; *Hanteren van gegevensmodel verbetert pakketselectie* (Use of data model improves software selection, only published in Dutch); Computable; April 1994

[HEI94.2] J.C.J. de Heij; *Onbegrip over referentie-gegevensmodellen: Objectieve onderbouwing van de juiste keuze* (Misunderstanding on reference data models: Objective foundation of the right choice, only published in Dutch); Computable; December 1994

[HEI94.3] J.C.J. de Heij; *The use of data models for assessing standard logistics software*; Computers in industry 25; 1994

[HOF93] J.D. Hofman and J.F. Rockart; *Using a template-based approach to systems delivery*; Center of Information Systems Research; 1993

[IDS94] IDS, Prof. Scheer; *Aris-Toolset Handbuch*; Version 2; 1994

[IPK94] FhG IPK Berlin; *System für Auswahl und Test von PPS-Systemen (SAT-PPS): Kriterienkatalog;* unpublished; 1994

[ITC95] Stichting ITC; *Onderzoek Shopfloor Control: gebruikerswensen en evaluatie SFC-systemen* (Research Shop Floor Control: user requirements and evaluation of SFC systems, only published in Dutch); 1995

[JON86] C.B. Jones; *Systematic software development using VDM*; Prentice-Hall; 1986

[KRA79] N.J.T.A. Kramer and J. de Smit; *Systeemdenken* (Systems thinking, only published in Dutch); Stenfert Kroeze; 1979

[LEE84] P.H. van der Lee; *Rapport Productie Besturings Pakketten* (Report Production Control Software, only published in Dutch); 1984

[LEE92] P.H. van der Lee; *Automatisering in voorraadbeheer en productiebesturing* (Computerisation in stock management and production control, only published in Dutch); Kluwer Bedrijfswetenschappen; 1992

[LOG86/94] Logiplan and Berenschot; *Softwarepakketten voor productiebesturing* (Software packages for production management, only published in Dutch); 1986 until 1994, revised every two years

[LOG94] P.H. van der Lee; *Vragenlijst processtructuren* (Questionnaire on process structures, only published in Dutch); unpublished; 1994

[LUN81] M. Lundeberg, G. Goldkuhl, A. Nilsson; *Information systems development - A systematic approach*; Prentice-Hall, Englewood Cliffs; 1981

[MAR86] J. Martin; *Information engineering: Introduction and Strategies & Analysis*; 1986

[MAR92] N. Martin and D. Hough; *The open systems revolution: Opportunities for the logistics industry*; Logistics Information Management, volume 5, no. 3; 1992

[MIL56] G.A. Miller; *The magical number, seven, plus or minus two: Some limitations on our capacity for processing information*; Psychological Review, Volume 63; 1956

[MIN92] H. Min; *Selection of software: The analytic hierarchy process*; International Journal of Physical Distribution and Logistics Management, Volume 22, no 1; 1992

[NIJS89] G.M. Nijssen, T.A. Halpin; *Conceptual schema and relational database design, A fact oriented approach*; Database design; 1989

[OBJ93] Object Technology; *Trends and expectations, a joint study*; OMG and Techvantage; July 1993

[ORL72] J.A. Orlicky, G.W. Plossi and O.W. Wight; *Structuring the bill of material for MRP*; Production and inventory management; Volume 13, no 4

[POR92] J. Porter; *Buying software that fits*; Systems 3X/400; July 1992

[RAY95] S.R. Ray and S. Wallace; *A production management information model for discrete manufacturing*; Production Planning & Control volume 6, no 1; 1995

[ROS77] D.T. Ross; *Structured analysis (SA): a language for communicating ideas*; IEEE Transaction Software Engineering; 1977

[ROS93.1] G. van Rossen; *Client/server: a new concept defined and evaluated* (only published in Dutch); Informatie volume 35, no 9; 1993

[ROS93.2] G. van Rossen; *Appearances of client/server systems* (only published in Dutch); Informatie volume 35, no 10; 1993

[SCH94] A.-W. Scheer; *Business process engineering: Reference models for industrial enterprises*; Springer-Verlag; 1994

[SLA92] A. Slater; *Major Information system selection*; Logistics Information Management; 1992

[SPU94] K. Spurr, P. Layzell, L. Jennison, N. Richards (and others); *Business objects: Software solutions*; John Wiley & Sons; 1994

[STA95] B. Standaert; *Pakketselectie met behulp van datamodellen* (Software selection with the help of data models, only published in Dutch); Business Logistics 2; 1995

[STE93] R.H.A. van Stekelenborg; *Inkoop en informatiesystemen: Een synergetische combinatie* (Purchasing and information systems: A synergetic combination, only published in Dutch); TUE, rapport R9302V2; March 1993

[TAY93] D. A. Taylor and P. Harmon; *Objects in Action: Commercial Applications of Object-Oriented Technologies*; Addison-Wesley Publishing Company; 1993

[TAY95] D.A. Taylor; *Business engineering with object technology*; John Wiley & Sons; 1995

[USA81] US Air Force; *Integrated computer-aided manufacturing (ICAM) architecture (IDEF0)*; Air Force Materials Laboratory; 1981

[VAR93] various; *CASE-tools: themanummer* (Special issue on CASE tools, only published in Dutch); Informatie no 35; 1993

[VEE91] E.A. van Veen; Modelling product structures by generic bills-of-material; Dissertation TUE; 1991

[VLM93] Th.M.J. van Rijn, B.V.P. Schyns et. al.; *MRP in Process: The applicability of MRP-II in the semi-process industry*; Vereniging Logistiek Management; Van Gorcum, Assen; 1993

[VON88] R. Vonk; *Analist Workbenches: een referentiekader* (Analist Workbenches: a reference framework, only published in Dutch); Informatie no 30, 1988

[WAD72] W. Wadsworth; *EDP Project Management Controls*; Englewood Cliffs (Prentice Hall); 1972

[WED90] T. Wedel; *Wissensbasierte dynamische Konfiguration der Parameter von Standardpaketten zur Productionsplanung und -steuerung am Beispiel des IBM-Modularprogramms COPICS: Grundkonzept und Einstellregeln*; Inaugural-Dissertation; Nurnberg; 1990

[WEE88] A.J. van Weele; *Inkoop in strategisch perspectief: Analyse, planning en praktijk* (Purchasing in a strategic perspective: Analysis, planning and practise, only published in Dutch); Samson; 1988.

[WEE93] H.J. Johansson, P. McHugh, A.J. Pendlebury, W.A. Wheeler III; *Business Process Reengineering: Breakpoint strategies for market dominance*; John Wiley & Sons; 1993

[WIJE93] G.M. Wijers; *CASE als visie* (CASE as vision, only published in Dutch); Informatie volume 35; 1993

Curriculum vitae

The author of this Ph.D. thesis was born on November 15, 1959 in The Hague. After completing his secondary education, Athenaeum B, he only studied Mathematics for a short time at the University of Leiden.

In 1979 he took a position as computer programmer at Caral Automatisering, and subsequently in 1980 he moved on to AKB (Consultants for Management Science and Software Development). There he worked in the area of operations research in combination with the development of business information systems. As developer, project manager, and consultant he developed and implemented various large-scale information systems.

In 1989 he started work at Coopers & Lybrand Management Consultants and is currently manager with the Manufacturing & Logistics department. In this position he is primarily active in the area of logistic management, business process re-engineering, and the selection and implementation of integrated logistic software systems.

Parallel to his daily work, in 1986 he started studying for a degree in Computing Science at Eindhoven University of Technology. He completed this degree in 1991 with the final research project 'Data Structures in Software Packages for Production Control'. He received the Mignot prize in 1992 in recognition of the quality of this research. The results of his research motivated him to continue this research at the Faculty of Technology Management culminating in this publication

STELLINGEN

behorende bij het proefschrift

Assessment of Business Information Systems

by Data Structures

Jos C.J. de Heij

I

De reikwijdte van het Referentie-DataModel mag niet gedefinieerd worden in termen van een branche, besturingsvorm of type productieproces, zolang de te vergelijken bedrijfsinformatiesystemen zich niet hiertoe beperken (zie hoofdstuk 4 van dit proefschrift).

II

Het Referentie-DataModel is geen model van een universeel bedrijfsinformatiesysteem maar kan wel gebruikt worden bij het genereren van specifieke informatiesystemen (zie hoofdstuk 4 van dit proefschrift).

III

Een adequate datastructuur is een noodzakelijke voorwaarde, maar geen voldoende voorwaarde om de functionaliteit van een bedrijfsinformatiesystemen te waarborgen (zie hoofdstuk 8 van dit proefschrift).

IV

Datastructuren beschrijven oplossingen en geen probleemstelling of gewenste functionaliteit (zie hoofdstuk 8 van dit proefschrift).

V

De structuren in het Referentie-DataModel zijn potentiële knock-out criteria bij de selectie van bedrijfsinformatiesystemen (zie hoofdstuk 8 van dit proefschrift).

VI

Een check-list streeft naar volledigheid (zonder inzicht), terwijl een referentiemodel streeft naar inzicht (zonder volledigheid).

VII

Zowel een check-list als een referentiemodel stimuleert het denken over ‘wat handig is’ en niet over ‘wat noodzakelijk is’.

VIII

Door de snelle (technologische) ontwikkelingen zijn de eigenschappen van bedrijfsinformatiesystemen, bewezen op grond van implementaties, per definitie verouderd.

IX

Een onderneming die kiest voor informatie technologie als 'competitive edge' kiest tevens voor 'continuous improvement' en 'continuous investment'.

X

In hoeverre object georiënteerde technologie leidt tot een revolutie in de ontwikkeling van informatiesystemen hangt af van de mate van standaardisatie van de semantische interactie tussen objecten en de acceptatie daarvan door software leveranciers.

XI

Een adviseur moet zichzelf terugverdienen en dient hierop afgerekend te worden (uitgaande dat ook *alle* resultaten, zoals vermindering van risico, financieel worden gekwantificeerd).

XII

De beslissing tot het volledig in eigen beheer ontwikkelen van een bedrijfsinformatiesysteem ten opzichte van het gebruik maken van standaardsoftware, is zelden gebaseerd op een integrale vergelijking van kosten en baten.

XIII

Afspreken om een afspraak te maken is een beleefde manier om geen afspraak te maken.

XIV

Eigenwijs zijn, is alleen willen leren van je eigen fouten.

XV

Er gaat veel tijd verloren met nadenken, wat meer voordenken zou veel fouten voorkomen en tijd besparen.