

2004

## A comparison of the relationship management methodology and the extended business rules diagram method

Jeff Campbell  
*Edith Cowan University*

Follow this and additional works at: [https://ro.ecu.edu.au/theses\\_hons](https://ro.ecu.edu.au/theses_hons)



Part of the [E-Commerce Commons](#)

---

### Recommended Citation

Campbell, J. (2004). *A comparison of the relationship management methodology and the extended business rules diagram method*. [https://ro.ecu.edu.au/theses\\_hons/143](https://ro.ecu.edu.au/theses_hons/143)

This Thesis is posted at Research Online.  
[https://ro.ecu.edu.au/theses\\_hons/143](https://ro.ecu.edu.au/theses_hons/143)

# Edith Cowan University

## Copyright Warning

You may print or download ONE copy of this document for the purpose of your own research or study.

The University does not authorize you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site.

You are reminded of the following:

- Copyright owners are entitled to take legal action against persons who infringe their copyright.
- A reproduction of material that is protected by copyright may be a copyright infringement.
- A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.

**A Comparison of the**

**Relationship Management Methodology**

**and the**

**Extended Business Rules Diagram**

**Method**

**Jeff Campbell**  
**Bachelor of Science (Computer Science with Honours)**

**This thesis is presented in fulfilment of the requirements for the degree of Bachelor of Science (Computer Science with Honours)**

**Faculty of Computing, Health and Science**  
**Edith Cowan University**

**October 2004**

## USE OF THESIS

The Use of Thesis statement is not included in this version of the thesis.

## **ABSTRACT**

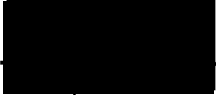
This research is a comparative study of the Relationship Management Methodology and the extended Business Rules Diagram method, when applied to the development of a Web-based hypermedia system. The RMM method focuses almost exclusively on the design phase of Web-based hypermedia systems with insufficient emphasis placed on the requirements analysis phase. The extended BRD method has been proposed to address this issue and attempts to cover more fully the development life cycle of Web-based hypermedia systems. A comparison of the main concepts, the phases of steps, as well as the modelling technique, notation and graphical representation is made between the two methods.

## DECLARATION

I certify that this thesis does not, to the best of my knowledge and belief:

- (i) incorporate without acknowledgment any material previously submitted for a degree or diploma in any institution of higher education.
- (ii) contain any material previously published or written by another person except where due reference is made in the text; or
- (iii) contain any defamatory material.

I also grant permission for the Library at Edith Cowan University to make duplicate copies of my thesis as required.

Signature: .....  .....

Date: ..... 8/2/2005 .....

## **ACKNOWLEDGEMENTS**

**I would like to thank:**

**Michael Johnstone my supervisor, who throughout the period of my honours year has provided me with immeasurable support and guidance.**

**My wonderful partner Merrilee, for providing me with the much needed support and encouragement to finish this research. You are a constant source of inspiration and my tower of strength.**

# TABLE OF CONTENTS

<b>USE OF THESIS</b> .....	<b>ii</b>
<b>ABSTRACT</b> .....	<b>iii</b>
<b>DECLARATION</b> .....	<b>iv</b>
<b>ACKNOWLEDGEMENTS</b> .....	<b>v</b>
<b>1. Introduction</b> .....	<b>1</b>
1.1 Background .....	1
1.2 The Significance of the Study .....	2
1.3 The Purpose of the Study .....	3
1.4 Research Questions .....	4
1.5 Glossary of Terms .....	4
<b>2. Literature Review</b> .....	<b>6</b>
2.1 General Literature on Current Hypermedia Development.....	6
2.2 Literature on Previous Findings .....	12
2.3 Specific Studies Similar to the Current Study.....	23
<b>3. Theoretical Framework</b> .....	<b>28</b>
<b>4. Case Study</b> .....	<b>32</b>
4.1 Department of Energy and Technology .....	32
4.2 The Web-based System.....	32
4.3 Development Strategy .....	33
<b>5. Data Collection and Case Study Preparation</b> .....	<b>35</b>
<b>6. The Extended Business Rules Diagram Method in Detail</b> .....	<b>37</b>
6.1 Identify Candidate Business Rules.....	37
6.2 Identify Candidate Events and Signals.....	38
6.3 Identify Candidate Objects .....	39
6.4 Construct Object Life Histories.....	39
6.5 Construct User Business Rules Diagram .....	40
6.6 Construct Business Rules Diagram.....	42
6.7 Construct Event Specification Table.....	43
6.8 Construct Process Design Diagram.....	43
6.9 Refine Event Specification Tables .....	46
6.10 Construct Navigation Design Diagrams.....	46
6.11 Construct Navigation Tables.....	47
<b>7. The Relationship Management Methodology (RMM) in Detail</b> .....	<b>48</b>
7.1 The Entity Relationship Diagram.....	48
7.2 Slice Design.....	49
7.3 Navigational Design .....	51
7.4 Conversion Protocol Design.....	53
7.5 User Interface Design.....	53
7.6 Runtime Behaviour Design .....	53
7.7 Construction and Testing .....	53
<b>8. The Extended Business Rules Diagram Method Applied to DET</b> .....	<b>54</b>
8.1 Identification of Candidate Business Rules .....	54
8.2 Identification of Candidate Events and Signals .....	55
8.3 Identification of Candidate Objects .....	56
8.4 Construction of Object Life Histories .....	56
8.5 Construction of User Business Rules Diagrams .....	57
8.6 Construction of Business Rules Diagram.....	58
8.7 Construction of Event Specification Tables.....	59



8.8	Construction of Processing Design Diagrams.....	60
8.9	Refine the Event Specification Tables .....	61
8.10	Construction of Navigation Design Diagrams .....	62
8.11	Construction of the Navigation Table .....	64
<b>9.</b>	<b>The Relationship Management Methodology Applied to DET.....</b>	<b>65</b>
9.1	The Entity Relationship Design .....	65
9.2	Slice Design.....	66
9.3	Navigation Design.....	67
9.4	Conversion Protocol Design.....	69
9.5	User-Interface Design .....	69
<b>10.</b>	<b>Analysis of Main Results.....</b>	<b>71</b>
10.1	Main Concepts of RMM .....	71
10.2	Main Concepts of the extended BRD Method .....	73
10.3	SDLC Phases Covered by the RMM and the extended BRD Method.....	75
10.4	Modelling Technique .....	75
10.5	Notation and Graphical Representation .....	77
<b>11.</b>	<b>Conclusions.....</b>	<b>79</b>
<b>12.</b>	<b>Further Research Work.....</b>	<b>82</b>
<b>13.</b>	<b>Concluding Remarks.....</b>	<b>83</b>
<b>14.</b>	<b>References.....</b>	<b>84</b>

# **1. Introduction**

## **1.1 Background**

In recent years the World Wide Web (WWW) has become the primary medium for successful communication for many organisations (Kyaw & Boldyreff, 1998). The WWW can provide clear and concise information and enable users' direct access to the content of an application. It becomes an even more effective medium through the use of hypertext and multimedia. To take advantage of this, organisations with development capabilities need to be placing a high priority on how they develop their Web-based hypermedia systems (Kyaw & Boldyreff, 1998).

It is important to distinguish a standard WWW application or page from a Web-based hypermedia system. A Web-based hypermedia system can be defined as an application that not only disseminates information, but also proactively interacts with a user to aid him/her in completing tasks (Takahashi & Liang, 1997). Furthermore, hypermedia systems attempt to emulate the intricate mechanisms of the human mind by associating blocks of knowledge with each other in a complex multitude of associative trails (Bush, 1945).

Although hypermedia pre-dates it, the WWW is only 14 years old (W3C, 2003) and as such the development of Web-based hypermedia systems is still an emerging discipline (Koch, 1999). Hypermedia systems that are implemented without following any real development methodology tend to confuse users and in the long run are more difficult to maintain (Koch, 1999). According to Kyaw and Boldyreff (1998), most organisations have produced WWW applications and IS without any structured and efficient design. They omit the user needs and application design requirements and as a result create maintenance problems.

Careful planning, design, standards and guidelines are essential infrastructure for the development of WWW based hypermedia systems, as is the case for conventional systems (Bieber and Vitali, 1997). One possible solution to address this unstructured approach to development of Web-based hypermedia systems would be to use development practices from related domains (software engineering, graphic design,

marketing etc.). However, there are suggestions that Web systems have unique characteristics that are at best poorly addressed by conventional development practices (Bieber and Vitali, 1997). Nanard and Nanard (1995) back this claim by stating that hypermedia development has to take into account aesthetic and cognitive aspects as well, that traditional software engineering environments do not support.

There have been a number of specialised methods proposed for the development of hypermedia systems. The most well reported of these methods are the Relationship Management Methodology (RMM) (Isakowitz et al., 1995), Hypermedia Design Method (HDM) (Garzotto et al., 1993), Object-Oriented Hypermedia Design Methodology (OOHDM) (Schwabe and Rossi, 1995), Structured Hypermedia Development Technique (SHDT) (Bichler and Nusser, n.d.) and the Scenario Based Object-Oriented Hypermedia Design Methodology (SOHDM) (Lee et al., 1999). The methods use different approaches and focus on different aspects, such as the design phase, but largely ignore any in-depth requirements analysis. Lowe (1999) asserts that it is assumed that the needs of the client are already well understood and hopefully well documented, though the fact that they are not is well demonstrated by the extent of problems over development scope which arise between web developers and their clients.

In 1998, McDermid proposed a requirements analysis approach called the Business Rules Diagram (BRD) method, which was extended by Johnstone (2004) to cover design, particularly in the area of hypermedia systems. The extended BRD method supports both the analysis and design of Web-based E-Commerce systems.

## **1.2 The Significance of the Study**

The WWW is a dynamic environment, fuelled by the continuous development of technologies and businesses that perceive a WWW presence as a way of gaining a competitive advantage. If methodologies, tools and technologies are to keep pace in this environment, then research to assess the current systems development practices within organisations would be expected (Wynekoop & Russo, 1997). Furthermore, if organisations are moving towards a greater reliance on Web-based IS (IS), the

adaptation and refinement of current software development methodologies needs to be an ongoing process. However, there is no empirical evidence that software development methodologies are keeping pace with technology or organisational change (Wynekoop & Russo, 1997), hence a similar argument may also apply to WWW applications. To further back this claim, one needs only to look at the number of Web sites that suffer from poor user interfaces and navigational problems to technological issues (Johnson and Nemetz, 1998).

The use of current software development methodologies and their application in developing Web-based hypermedia systems has largely been unevaluated. Wynekoop & Russo (1993) propose that IS development research assumes that the use of software development methodologies is effective, but such claims have not been validated. Therefore in the context of the WWW applications, it cannot be stated conclusively that the use of these methodologies will support future Web-based hypermedia systems.

Developers of Web-based hypermedia systems need to gain an understanding of the development process. This can best be achieved by evaluating the current methodologies and finding out what is needed before suggesting new advances in methodologies, tools and techniques (Paynter & Pearson, 1998). This research will provide insights into the effectiveness of the use of current software development methodologies, in particular the RMM method and the extended BRD method, when applied to the development of a Web-based hypermedia system.

### **1.3 The Purpose of the Study**

The purpose of this research is to evaluate the effectiveness of current Web-based hypermedia development methodologies. One commonly used development method is the Relationship Management Methodology (RMM), which is based on the E-R model and views hypermedia as a vehicle for managing relationships among information objects (Isakowitz et al., 1995). It has been suggested however that the most commonly used methods for developing Web-based hypermedia systems, including RMM, tend to focus on the design aspect of hypermedia systems and do not

properly address any requirements elicitation or analysis (Johnstone et al., 2003). To address this deficiency, Johnstone et al. (2003) proposed an extension to McDermid's (1998) Business Rules Diagram (BRD) method, which supports both the analysis and design of Web-based hypermedia systems.

A comparison will be made between the RMM methodology and the extended Business Rules Diagram (BRD) method. The research will look at the similarities and differences between the two methods when applied to the development of a Web-based hypermedia system and their effectiveness in addressing the future needs of such systems. The study will also attempt to validate the claim that the extended BRD method covers more fully the development life-cycle of web-based hypermedia systems.

#### **1.4 Research Questions**

How does the Relationship Management Methodology (RMM) compare to the extended Business Rules Diagram (BRD) Method, when applied to the development of a Web-based hypermedia system?

#### **1.5 Glossary of Terms**

**E-R Model:** An approach to data modelling proposed by Chen (1976). The model says that you divide your database in two logical parts, entities (e.g. "customer", "product") and relations ("buys", "pays for"). Entity-relationship diagrams can be used to represent a model.

**Hypertext:** A special type of database system, invented by Ted Nelson in the 1960s, in which objects (text, pictures, music, programs, and so on) can be creatively linked to each other. When you select an object, you can see all the other objects that are linked to it. You can move from one object to another even though they might have very different forms (Webopedia, 2003).

**Hypermedia:** Applications that allow users to forge their own non-linear paths through images, sound and text (Berk & Devlin, 1991, p23).

**Multimedia:** The use of computers to present text, graphics, video, animation, and sound in an integrated way (Webopedia, 2003).

**Relationship Management Methodology (RMM):** A methodology that provides a structured design for hypermedia applications. The focus is on modelling the underlying content, the user viewpoints onto this content and the navigational structures that interlink the content (Lowe & Henderson-Sellers, n.d.).

**Business Rules Diagram (BRD) Method:** A process oriented, state based requirements analysis method. It is process oriented, as the information the BRD method captures is dynamic. The system or one of its components that the BRD method describes can be in one of a number of states, hence the description state based. As described by Johnstone et al. (2002), the BRD method depicts states, events, conditions, signals and blobs.

- States reflect the status of a system or one of its components.
- Events are actions carried out internally by the organisation.
- Conditions define the criteria by which objects of interest in the business move from one state to the next as events take place and are sometimes known as "if-then rules" in other systems.
- Signals (arrows) either enter or leave the human activity system.
- The Harel blob (Harel, 1988) encapsulates other constructs and is used to model selection or simultaneous action.

## 2. Literature Review

### 2.1 General Literature on Current Hypermedia Development

To appreciate and understand the evolution of hypertext/hypermedia research, a brief look at the major events in its history is beneficial. Dr Vannevar Bush published an article in 1945 and in that article he proposed a machine named Memex that would have the capacity to store textual and graphical information in such a way that any piece of information could be arbitrarily linked to any other piece of information (Feizabadi, 1998). The essential feature of Memex was the process of tying two information items together which would give a user the ability to create an information trail of links traversed, which if required, could be retrieved at a later date (Bush, 1945).

This description of a machine that would allow a non-sequential form of composing and writing could best be described as the conception of hypertext (Nielsen, 1990, p 41), although due to technological constraints at the time, a working version could not be fully realised (Bush, 1945). Bush understood this but considered Memex to be an immediate step towards associative indexing, the basic idea of which is a provision whereby any item may be caused at will to select immediately and automatically another (Bush, 1945).

In the following years, the field of hypertext research continued to mature as Bush's initial vision was expanded. In 1965, Ted Nelson coined the terms hypertext and hypermedia in a paper to the ACM 20th national conference. Nelson (1990, p2) defined hypertext as meaning "*non-sequential writing or text that branches and allows choice to the reader, best read at an interactive screen*" with the term hypermedia being used to describe a system or document whose focus is on non textual types of information (Nielsen, 1988).

Using the work of Nelson and others as a base, a team of researchers led by Andries van Dam developed the first hypertext-based system in 1967. IBM funded the research and the first hypertext implementation, *Hypertext Editing System*, ran on an

IBM/360 mainframe. IBM later sold the system to the Houston Manned Spacecraft Centre which reportedly used it for the Apollo space program documentation (Feizabadi, 1998).

The next major landmark in the hypertext time scale was conceived by Nelson in 1967. Nelson's continuing interest in all things related to hypertext became the Xanadu project. The Xanadu project has been in development for more than 30 years and was owned for a while by Autodesk, but was later dropped (W3C, n.d). Gromov (2003) describes the main features of the Xanadu project as:

- the transclusion feature allows quotation of fragments of any size with royalty to the original publisher;
- a system for the network sale of documents with automatic royalty on every byte;
- an implementation of a connected literature;
- a system for a point-and-click universe; and
- a completely interactive docuverse.

While the Xanadu project has never been fully realised in a commercial sense, Nelson's research has inspired a number of followers including computer programmers, managers and executives who continue to fund and conduct hypertext research (Feizabadi, 1998).

Hypertext research continued during the 1970s and it was not until 1978 that what is argued to be the first true hypermedia system called the Aspen Movie Map was developed (Naimark, 1997). The Aspen Movie Map was a surrogate travel application that allowed the user to enjoy a simulated ride through the city of Aspen. It was not really designed to help people achieve specific tasks, but rather was an attempt to show what was possible with the technology at the time. Even today the Aspen Movie Map remains one of the most sophisticated hypermedia systems ever built (Interactivity in Motion, 2002).



The history of hypertext research does not stop there, but new developments since the 1980s are too numerous to list. Neilsen (1990, p41) says that hypertext was conceived in 1945, born in the 1960s, slowly nurtured in the 1970s, entered the real world in the 1980s with an especially rapid growth after 1985, culminating in a fully established field during 1989. In recent years, hypertext and hypermedia concepts have increased in popularity and this has largely been due to the success of the World Wide Web (WWW) and the ability to disseminate information to a larger audience (Nykanen, n.d.). A Hypertext timeline from 1945 to 1989 is shown in table 2.1.

**Table 2.1 A Timeline of the Major Events in the History of Hypertext**

<b>Year</b>	<b>Development</b>
1945	Bush proposes Memex in "As We May Think"
1965	Nelson coins the term Hypertext
1967	Andries van Dam develops the Hypertext Editing System
1978	A team at MIT, headed by Andrew Lippman, develops the Aspen Movie Map
1979	Ted Nelson begins the Xanadu project
1985	Janet Walker develops the Symbolics Document Examiner, the first hypertext system used by real customers
1986	OWL introduces Guide for Macintosh, based on the Unix Guide system, developed by Peter Brown which allows text and graphics
1987	Apple delivers HyperCard free with every Macintosh, the first widely available personal hypermedia authoring system
1989	Tim Berners-Lee proposes the World Wide Web (WWW)

The concept of the WWW began in 1980 when Tim Berners-Lee, while working at the Centre Européen de Recherche Nucléaire (CERN), got frustrated with the fact that data he was working on was stored and manipulated on several different databases (Feizabadi, 1998). This made combining data virtually impossible as the databases were isolated and had practically no interaction or connectivity. As a result, Berners-Lee set out to develop a system that would allow him to store random associations between arbitrary pieces of information (Feizabadi, 1998).

His first attempt at solving this problem was a program he wrote called “Enquire-Within-Upon-Everything”. After completing the ENQUIRE system, Berners-Lee left CERN and while the system went largely unused it was the catalyst for further work that he would conduct when he returned to CERN in 1989 (W3C, 2000). In 1989 Berners-Lee submitted a paper titled “Information Management: A Proposal” that described a system that would allow fellows, students and visiting scientists the ability to access information that was geographically dispersed in a quick and efficient manner (Feizabadi, 1998).

The proposal built on the idea of Bush’s hypertext based system Memex and called for a method of combining several different servers of machine stored information. This would provide users with a single interface to a large repository of data, ranging from reports, notes, databases, computer documentation and on-line help (Feizabadi, 1998).

Relihan et al. (1994) outlines the proposal's main objectives:

- to provide a simple protocol for requesting human readable information stored in remote systems accessible using networks;
- to provide a protocol by which information could automatically be exchanged in a format common to the information supplier and the information consumer;
- to provide some method of reading text (and possibly graphics) using a large proportion of the display technology in use at CERN at that time;
- the provision and maintenance of collections of documents, into which users could place documents of their own;
- to allow documents or collections of documents managed by individuals to be linked by hyperlinks to other documents or collections of documents;
- the provision of a search option, to allow information to be automatically searched for by keywords, in addition to being navigated to by the following of hyperlinks;

- to use public domain software wherever possible and to interface to existing proprietary systems; and
- to provide the necessary software free of charge.

As described by Feizabadi (1998, p15) Berners-Lee divided the project into two phases:

*"In the first phase CERN would "make use of existing software and hardware as well as implementing simple browsers for the user's workstations, based on an analysis of the requirements for information access needs by experiments." In the second phase of the project they would "extend the application area by also allowing the users to add new material."*

While initially the project did not receive full support, it was recirculated in 1990 and in October of that year work began on the project and it was re-named the World Wide Web (W3C, 2000). The World Wide Web (WWW) programme paved the way for Web development in the future by introducing their server and browser; the Hypertext Transfer Protocol (HTTP) which is used for the communication between the clients and the server; the Hypertext Markup Language which is used to compose Web documents; and the Universal Resource Locator (URL) which is the global address of documents and other resources on the WWW (Feizabadi, 1998).

The WWW has become such an integral part of the development of hypermedia systems that it simply cannot be ignored (Nykanen, n.d.). It has taken the basic hypertext/hypermedia paradigm, where information is interconnected by links and users can access those information items by navigating through this non-linear structure, and made this model the standard way of accessing information on the Internet (Dolog et al., 2003).

The way in which information is presented on the Internet varies from small self-constructed Web pages to large complex Web systems and consequently there is a difference between developing a set of homepages for a small Web-site and

developing systems on the Web that manage information, interact with databases and are integrated with other enterprise IS. The latter, called Web system development requires well thought-out methods and different tools than those required for homepage creation (Isakowitz, 1997).

Although there has been a multitude of innovative software tools developed to support various aspects of hypermedia development, the generated hypermedia systems still suffer from problems which range from simple aspects such as poorly designed user interfaces and navigation, to misuse of the available technology (Johnson and Nemetz, 1998). Some developers (see, for example, Koch, 1999 and Lang, 2002) lay the blame for these problems on the use of ad hoc approaches to Web-based hypermedia systems development, while others such as Linden and Cybulski (2003) imply that there is a widespread lack of knowledge and/or experience in hypermedia systems design and evaluation of multimedia development results.

With the growth, size, complexity and heterogeneity of current hypermedia systems such as the WWW, it becomes a difficult task to encourage developers to follow guidelines concerning the overall organisation of hypermedia information. Other reasons may include lack of training/education, the apparent ease of construction of Web-based hypermedia systems and the commercial pressures on developers, hence the use of ad hoc approaches for systems development. This is particularly true for Web-based applications, which are expected to be used by a much greater variety of users than any earlier standalone application (Brusliovsky, 1998, p30).

One of the problems that the literature related to hypertext/hypermedia development has identified, is that Web-based hypermedia systems are sometimes categorised as a form of software system and therefore have been developed using traditional IS approaches (Lowe and Henderson-Sellers, n.d.). While the use of traditional software tools and methodologies may suit some Web-based systems development, there is growing evidence in the literature to suggest that a different approach needs to be adopted to cater for the unique characteristics of Web-based systems (Koch, 1999,

Lang, 2002, Barna et al., n.d, Costagliola et al., n.d, Lowe and Henderson-Sellers, n.d).

## 2.2 Literature on Previous Findings

There are a number of researchers who support the notion that hypermedia systems design presents challenges that are not normally encountered with the development of traditional IS. Lang (2002) has stated that these challenges and the view that conventional software engineering techniques provide inadequate solutions, has resulted in support for specialised methods to be applied to the development of Web-based hypermedia systems. Lang and other researchers have explored and documented how Web-based hypermedia systems differ from traditional IS development to support this claim.

Eklund and Lowe (2002) state that the research literature detailing the differences between Web systems and more conventional software systems is steadily growing and identify unique characteristics of these systems reflecting technical, usability and organisational issues. These characteristics include aspects such as a tighter link between the business goals and the system architecture, a complex information architecture (Russell, 2000), increased importance of quality attributes as applications more often than not represent the interface between the user and the organisation, rapidly changing technologies and open modularised architectures (Eklund & Lowe, 2002).

Lang (2002) also identified a number of areas in which Web-based hypermedia systems differ from conventional systems. Based on Bush's description of hypermedia as "*trying to emulate the intricate mechanisms of the human mind, by associating blocks of knowledge with each other in a complex multitude of associative trails*" (Bush, 1945, p101), Lang states that hypermedia systems tend to be more complex than conventional systems because the implementations of such systems lends itself to the creation of 'spaghetti like' structures. Even in organisations with small systems the problem of storing, processing and presenting data becomes a major

issue, as IS are no longer just handling simple text and numeric data types but multimedia data, which is always evolving and becoming richer in content.

Another difference that Lang (2002) describes is also identified by Koch (1999), where people with very different skills are involved in the development process at different stages. They range from authors, layout designers, programmers, multimedia experts and marketing specialists, often with very little in common, making it much more difficult to develop hypermedia systems using traditional methods. Whilst this type of skills diversity is not a phenomenon confined to hypermedia development, participants of Web-based systems development tend to hail from disciplines that are not just diverse but also discrete, such as systems analysis, graphic design, marketing and media production. These disciplines each have their own very different approaches to systems development, which often results in a lack of communication between sections and cultural conflicts (Gallagher & Webb, 1997).

Product development life cycles is another difference that has been identified. Web-based systems can now be deployed with such ease and speed, that the development life cycle has accelerated from that reported by Fitzgerald (1997) who states that the timescale for typical IS (IS) development projects spans less than six months from the time of initial requirements capture to delivery of the project, with project teams of no more than three developers. He further states that this development life cycle reflects the profile of a small-scale rapid development. With hypermedia development often occurring within a much shorter time frame, Thomas characterises Web-based application development as "*guerrilla programming in a hostile environment using unproven tools, processes and technology*" (Thomas, 1998, p79).

Lang (2002) also identifies a functional difference between IS of the past and current Web-based systems. IS were traditionally designed to carry out internal functions within organisations, however with the advent of the WWW, Web-based systems are being developed with more of an external focus as they tend to serve as the interface between organisations and the environments in which they operate. Because of this

external focus, the usability of Web-based systems becomes a critical issue (Buckingham, Shum and McKnight, 1997). The end user must be able to navigate the system with ease to achieve a desired result. Furthermore, unlike traditional systems, there is no pre-defined category of user from which requirements emanate, making it difficult to tailor a solution to a specific set of users which was previously done with traditional IS (Lang, 2002).

The last difference Lang (2002) identifies is in the area of ongoing maintenance with respect to content. The WWW is a dynamic environment and as such the content of Web-based systems tends to be inherently dynamic as well. For the content of such systems to be relevant and current, maintainability tends to be crucial and involve significant overheads. The evolution of Web-based systems content tends to be more granular and undergo much more continuous refinement than traditional systems, therefore a greater importance needs to be placed on future requirements as systems grow and this growth needs to occur in a controlled but flexible manner.

Even though Web-based systems can be viewed as software systems, the current literature supports the notion that it would be inappropriate to apply existing representations of software development methodologies to Web system models. To do so may encourage developers to overlook the unique requirements of these Web systems, leading to inappropriate solutions (Eklund & Lowe, 2002). If existing models are used, developers may need to exercise caution in their application of these methods to the development of Web systems and may need to be constantly aware of the limitations with respect to aspects of the Web systems they wish to understand and document (Lang, 2002).

With this in mind, there is a need to identify the problems/deficiencies with the methods that are currently being used to develop Web-based hypermedia systems. There is a clear division on views here as to whether there are any problems and if new methods are required at all. While much of the current literature has identified deficiencies in hypermedia development methods and stated that a transposition of traditional IS development methods is both difficult and inadequate (Nanard and

Nanard, 1995), Keen counters that argument by stating that, “*in practice, very few of the major areas of concern in IS Research are less than twenty years old and issues seen as new often turn out to have long roots*” (Keen, 1991, p 30).

Another viewpoint expressed by Lowe et al. (1999) suggests that the process by which hypermedia applications can be developed has been given little consideration and consequently, certain requirements or characteristics tend to be omitted. In addition, the work that has been carried out, such as the work on methods (e.g. Relationship Management Methodology), has tended to be prescriptive and has not really provided any significant indication of how the methods can be readily adapted to different application domains. Table 2.2 lists some of the better-known methods that have been proposed for the development of Web-based hypermedia systems.

**Table 2.2 - Development Methods for Web-based Hypermedia Systems, taken from Lang (2002)**

<b>Method</b>	<b>Author(s)/Developer(s)</b>
Relationship Management Methodology (RMM)	Isakowitz et al. (1995)
Hypermedia Design Method (HDM)	Garzotto et al. (1993)
Object-Oriented Hypermedia Design Methodology (OOHDM)	Schwabe & Rossi (1995)
World Wide Web Design Technique (W3DT/SHDT)	Bichler & Nusser (1996)
Scenario Based Object-Oriented Hypermedia Design Methodology (SODHM)	Lee et al. (1999)



In an attempt to understand the possible deficiencies inherent in the current methods used to develop Web-based hypermedia systems, a brief description of the methods listed in table 2.2 is appropriate.

The Hypermedia Design Method (HDM) is one of the first methods that was developed to define the structure and interaction in hypermedia applications (Garzotto et al., 1993). It is based on the Entity Relationship (E-R) and database design techniques and is concerned with how the HDM entities are linked, how navigation is performed within those links and how that information is visualised. Garzotto et al. (1993) describe HDM entities as information chunks that can end up being quite large structures, which are representations of some real world object of the application domain. The difference between HDM and other modelling methods is that it is aimed at modelling applications rather than systems (Garzotto et al., 1993). As such it only covers the conceptual, navigational and presentational design stages of the development lifecycle with no real emphasis on requirements capture, analysis, implementation or testing stages.

The Relationship Management Methodology (RMM) is also based on the E-R model with the addition of the concept of slices, which are presentation units and appear as pages of hypermedia applications (Koch, 1999). RMM, described later in more detail, focuses on the design stage of the development lifecycle and in particular how the content is modelled, the viewpoint of the user and the navigational structures that interlink the content, with little or no requirements analysis (Johnstone et al., 2003).

The Object-Oriented Hypermedia Design Method (OOHDM), as explained by Schwabe and Rossi (1996), covers the following four phases in its development lifecycle:

- conceptual modelling;
- navigational design;
- abstract interface design; and
- implementation.

This four step process supports an incremental or prototype process model. While there are some requirements gathering, through scenario and use case analysis, interviews and the mapping of user interface diagrams to the conceptual model, the focus is almost exclusively on the design phase in terms of the navigation and interface design aspects of the logical structure of the application domain (Kyaw & Boldyreff, 1998).

The Structured Hypermedia Development Technique (SHDT) method comprises information structuring, navigational design, organisational design, interface design, implementation and introduction and maintenance (Kyaw & Boldyreff, 1998). Again the focus is on the design phase with little emphasis on any requirements capture, analysis or testing.

The Scenario Based Object-Oriented Hypermedia Design Method (SOHDM) is a scenario-based methodology, which consists of the following stages:

- domain analysis;
- OO modelling;
- view design;
- navigational design;
- implementation design; and
- construction.

As described by Koch, *“SOHDM differs from other methods in that it uses what are called scenarios, which are described through scenario activity charts that are based on events activities and activity flows”* (Koch, 1999, p6). However, it adopts a similar approach to the previously described methods, in that a large part of the development lifecycle is spent in the design phase, particularly the navigation design, which uses scenarios to determine access structure nodes.

From the brief descriptions above, it is apparent that most of the methods have a common thread in that the primary focus is on navigation and design and little emphasis is placed on analysis or requirements capture. In addition, each of the methods has failed to adequately detail the process by which these systems can be developed (Johnstone et al., 2003).

Even if these deficiencies were to be ignored, a recent study has found that none of these methods are being used to any significant extent by Web-based systems developers (Barry and Lang, 2001). A reason for this could be that most of the literature on hypermedia systems development lacks sufficient detail on the design process and focuses either on high-level corporate planning or low-level technical implementation. Lang (2001, p4) states that:

*“There is little in the way of either theory or empirical field research to guide developers in practical applications of Web-based hypermedia systems development. What few guidelines that do exist tend to be informal and are in the main biased towards human-computer interaction issues”.*

In order to understand how and why developers use specific software development methodologies for Web-based hypermedia systems, one must first have a good understanding of the basic concepts involved. One problem that developers of hypermedia systems face is deciding which aspects of the development cycle need to be covered by a methodology. Rumbaugh (1995) has suggested that a method should include the following components:

- a set of modelling concepts to capture semantic knowledge about the problem and its solution;
- a set of graphical tools and notations for presenting the underlying modelling information;
- a series of steps, iterative in nature, for constructing models and implementations of them; and

- development hints and rules of thumb that can be applied during the development process.

Henderson-Sellers (1995) suggest a more extensive list of aspects that have to be covered by a methodology, namely:

- a full life-cycle process;
- a full set of concepts and models that are internally self-consistent;
- a collection of rules and guidelines;
- a full description of deliverables;
- a workable notation;
- a set of metrics, together with advice on quality, standards and test strategies;
- guidelines for project management;
- advice for library management and reuse; and
- identification of organisational rules.

By incorporating aspects of this list and using a method in preference to an ad hoc approach, developers of Web-based hypermedia systems derive a number of benefits as described by Kyaw and Boldyreff (1998). Developers are able to design large complex systems that contain data with intricate relationships. Invariably, project teams are also helped by having detailed descriptions of the process and steps required to complete a project. The separation of each phase assists teams in assessing progress and addressing development issues. The support for techniques used in design methods (such as Object-Oriented (OO), E-R etc.), can sometimes help developers solve large and complex design problems in various application domains and the use of a design method enables developers to implement solutions in various ways (i.e. HTML, XML etc.).

However, Kyaw and Boldyreff (1998) point out that the use of a method also has its limitations. Sometimes the complexity of the method itself necessitates the need for developers who are already skilled in various design techniques. In addition, the time

and effort required initially to develop a system using a design method is substantially more than if an ad hoc approach was used, however maintenance of these systems becomes a huge problem particularly in the area of cost. And lastly, because research in hypermedia design is still an emerging discipline, there is little support for CASE tools based on these methods.

As described previously, there are already a number of specialised methods that have been developed to address the issue of how best to implement hypermedia systems for the WWW, however, as the literature has shown, when applying a new technology the resulting efforts are often modelled on old technologies. Many of the current methodologies tend to cover only certain parts of the development life cycle, predominantly focussing on the design phase of these systems (Koch, 1999).

RMM, developed by Isakowitz et al. (1995), is one such method and was developed with the following aims in mind:

- to facilitate the design of Web-based hypermedia systems and their seamless integration with databases and enterprise wide IS;
- RMM applies best to applications that are large and dynamic, where data is stored in databases, and where hyper-links are computer generated; and
- the design of a Web-based hypermedia system can be succinctly represented with RMM diagrams so that designers, programmers and users have a clear understanding of the scope and structure of the application.

RMM proposes a series of seven steps as detailed by Isakowitz et al. (1995) and are as follows:

- **Step 1: E-R Design** – An entity relationship diagram is constructed of the information domain of the application.
- **Step 2: Slice Design** – This step is used to determine how the information in the chosen entities will be presented to the users and the access they have to those entities.

- **Step 3: Navigational Design** – Associative relationships appearing in the E-R diagram are analysed and paths between those entities are designed to enable navigation of the hypertext.
- **Step 4: Protocol Conversion Design** – Converting design components into physical objects.
- **Step 5: User Interface Design** - Concerned with the design of screen layouts for every object.
- **Step 6: Run Time Behaviour Design** – concerned with the decisions about how link traversal, history, backtracking and navigational mechanisms are to be implemented.
- **Step 7: Construction and Testing** – Navigational paths are thoroughly tested.

One of the purported advantages of using RMM is that it provides a clear series of steps, thereby providing a structured approach to the design. This structured approach allows developers to eliminate the 'spaghetti like' code that Isakowitz et al. (1995) describe and present users with well-structured sites that are easy to navigate and maintain. Another advantage is that different groups can work at the same time on different design aspects. This is because the design phase is further separated into additional phases, each focussing on a particular aspect. For example, one group can be working on the navigational design while another group can be working concurrently on the user-interface design. In addition, RMM developed Web systems are easily updated as data updates can be submitted either directly through the Web, or via database APIs or SQL queries (Isakowitz et al., 1995).

One of the major disadvantages of the RMM methodology as Johnstone et al. (2003) point out is that requirements analysis is largely ignored and the focus is exclusively on design. Isakowitz et al. (1995) also state that RMM focuses on the design, development and construction phases, with little attention paid to analysis. As such, the RMM method does not really cover the entire development lifecycle.

In 1998, McDermid proposed an approach to modelling system requirements called the Business Rules Diagram (BRD) method. The BRD method as described by

Johnstone et al. (2002), *“utilises a state-based model which has a notation similar to, but more powerful than, flowcharts”*. McDermid (1998) lists seven steps required to perform a complete BRD analysis. They are as follows:

- identify candidate business rules;
- identify candidate events and signals;
- identify candidate objects;
- construct Object Life Histories (OLHs);
- construct User Business Rule Diagrams (UBRDs);
- construct Business Rules Diagrams; and
- construct Event Specification Tables (ESTs).

The BRD method provides developers of IS with a potentially effective tool for capturing requirements. Given this, Johnstone et al. (2003) determined that it was appropriate to extend the BRD method to address the deficiencies inherent in many of the more common hypermedia system development methodologies. Firstly, the extensions proposed involve separating the navigation logic from the processing logic in the BRD diagrams. Secondly, the BRD notation was refined to ensure that the navigation specific symbols mirrored the processing specific symbols already present in the BRD notation. The symbols chosen to represent the design notation were not representative of any specific technology because of the high turnover rate in the use of these technologies in e-commerce systems. A more generic symbol notation was developed that could be applied to the design of any type of Web-based e-commerce system.

Put more succinctly, Johnstone et al. (2003) incorporated four additional steps to McDermid's (1998) original BRD analysis method steps as follows:

- construct Process Design Diagrams (PDDs);
- refine Event Specification Tables (ESTs);
- construct Navigation Design Diagrams (NDDs); and





There have been a number of comparative studies undertaken that have looked at development methodologies for Web-based hypermedia systems such as studies conducted by Koch (1999) and Kyaw & Boldyreff (1998). Koch's study examined eleven of the better-reported hypermedia development methods. Each method was described briefly and then three separate comparative studies were conducted, each focussing on different aspects. The studies as detailed by Koch (1999) were as follows:

1. The first comparative study focussed on the steps to be performed when using a particular method, the modelling technique used as well as the graphical representation and notation chosen for the development. Koch also investigated the support for CASE tool support available for these development methods.
2. The second comparative study examined the design concepts used by each method and looked at the conceptual, structural and presentation levels of each method (i.e. relationships, how those relationships are linked and how users are presented with this information).
3. The third comparative study looked at the phases that each method covered in the development lifecycle. A comparison of the depth of description and guidelines provided by each method, to assist in the development of a hypermedia system was made.

The main aim of Koch's study was not so much to criticise any one particular method, but to identify the strengths and weaknesses of each method when applied to a particular issue in the development lifecycle. The benefit here is that developers would have a better understanding of when to use a particular development methodology that would suit their project goals, or if a particular method was used, where in the development life cycle alternative strategies needed to be employed to address weaknesses of that method.

Koch (1999) concluded that the evolution of hypermedia development is following a similar evolutionary path to that of software development and that the field is

continuing to evolve and improve. By identifying the strengths and weaknesses of each method, Koch believes it will help other researchers in their comprehension of the current methodologies being utilised and help to steer research in areas where it is required. Particular aspects that needed improvement were in the areas of requirements capture, validation, verification and testing of each method so that it covered more fully the development life cycle.

Kyaw and Boldyreff's study (Kyaw and Boldyreff, 1998) looked at nine criteria they considered should be covered by a hypermedia design method. The criteria they detailed are as follows:

- **Notations of design:** Notations of different design approaches in each development stage (e.g. such as E-R diagrams, Object Oriented class diagrams, etc). Notations include the representation of the application domain from very general top view to all the attributes and the relationships among them.
- **Multimedia support:** The adequate support for accessing, presenting and navigating multimedia data for hypermedia application domain.
- **Modelling constructs:** The consideration of data modelling constructs by each design methods. This includes conceptual and navigational design constructs and interface design activities. It also consists of modelling approaches for static and dynamic navigation.
- **Database support:** The support of hypermedia access and navigation structure for building hypermedia database application domain adequately.
- **Reusability:** It includes the reusability of pre-existing classes, models and design constructs. For example, how well a design method can support reusability to develop new application classes using existing classes.
- **Maintenance:** Maintenance of hypermedia applications is a time taking and difficult task since they are constructed using complex designs and not directly accessible at run time. An evaluation will be made on how efficient it is to maintain and update the applications developed using different design methods.
- **Types of applications:** Since there are broad ranges of hypermedia applications, there is no formal design approach that is relevant to designing

all of them. Evaluation will also be made on which design methods are most suited to different types of applications.

- **User Interface Design:** The support of user interface design, including GUI techniques and other widgets.
- **Tools support:** The use of CASE tools to support each method.

Using these criteria as a comparative baseline, Kyaw and Boldyreff go on to examine three main hypermedia design methods (i.e. HDM, OOHDM and RMM). Their study gives a brief description of each method, much like Koch's and the three methods are then compared also taking into account:

- the levels of complexity each method covers;
- how the method is applied;
- the goals of each stage of the development lifecycle; and
- what deliverables and diagrams will be produced by each method.

Kyaw and Boldyreff's conclusions revolve around their identification of the strengths and weaknesses of each method with respect to the nine criteria they identified as being essential for a hypermedia design method to cover. The main aim of their study is similar to Koch's in that they see their contributions as assisting designers and developers in choosing the right tools and methods for a specific application (Kyaw & Boldyreff, 1998). They further suggest that as more and more methods are developed, there will be an increased need for standardisation across all methods in the areas of notations, modelling and implementation.

Based on these two studies, comparisons of the extended BRD method and the RMM method will closely follow Koch's study and be limited to:

- a comparison of the steps to be performed when using a method, the modelling technique used as well as the graphical representation and notation chosen for the development;
- a comparison of the design concepts used; and
- a comparison of the phases of the development life cycle covered by both methods.

It is expected that this research will validate the claim that the extended BRD method covers more fully the lifecycle of Web-based hypermedia systems development. It will also identify the strengths and weaknesses of both the RMM method and the extended BRD method, which will assist future research in determining the suitability for applying each method to a hypermedia development project.

### 3. Theoretical Framework

There has been an increase in the use of case study research in the field of IS for at least two decades (Benbasat et al., 1987; Lee 1989; Orlikowski and Baroudi, 1991; Alavi and Carlson 1992; Yin 1993; Markus 1997; Klein and Myers 1999). A number of researchers such as Robey and Newman (1996), Pare and Elam (1997) and Sarker and Lee (2000) have used case studies to study various IS phenomena. The case methodology is appropriate when there is not a strong theoretical base supporting the phenomenon under investigation (Paynter and Pearson, 1998).

The research approach to be used is case study research and is applicable for the following reasons:

- *the study of the design of WWW-based IS cannot be easily studied outside its natural development environment;*
- *the research question directly addresses a contemporary event i.e. the analysis and design of IS for the WWW; and*
- *WWW-based IS and their design do not enjoy an established theoretical base* (Paynter and Pearson, 1998).

A case study strategy is also appropriate as it is being used as an evaluation tool. Schramm (1971) states that the essence of a case study or that which is common among all types of case study is that it tries to make more clear the decisions that were made, why they were made, how those decisions were implemented and what occurred as a result of those decisions.

Yin (2003, p13) defines the scope of a case study as "*an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident*". The methodology to be used is that outlined by Yin (2003, p21) and consists of four stages:

1. design the case study;
2. conduct the case study;
3. analyse the case study evidence; and
4. develop the conclusions, recommendations and implications.

The scope of this case study research is largely technical i.e. it will involve the technical and functional aspects of the Web-based hypermedia system to be developed, with which all staff interact and utilise, not only the developers. The context specifically includes how the system to be developed fits into the routine day-to-day operating environment; geographic and physical limitations, matters of style, culture and values unique to the specific industry.

Yin (2003, p13) presents at least four applications for a case study model:

1. to explain complex causal links in real-life interventions;
2. to describe the real-life context in which the intervention has occurred;
3. to describe the intervention itself; and
4. to explore those situations in which the intervention being evaluated has no clear set of outcomes.

Tellis (1997) states that Information Technologies involve all four of the categories listed above and because of this the case study method is considered appropriate for this research. The research will also employ a single case rather than a multiple case design. The rationale for this approach is that a single case can be used as a critical test of a significant theory (Grosse et al., 1971). Yin (2003, p40) also agrees that a single case can be used to confirm a theory's propositions or reveal alternative explanations that might be more appropriate. Yin (2003, p40) goes on to state that the single case can make a significant contribution to the existing knowledge base and theory. By using a single case, researchers are able to conduct an in-depth investigation to provide a rich description and understanding (Darke & Shanks, 2000, p97).

There has been some criticism that single case studies cannot be used to generalise about the broader class. Dogan and Pelassey (1990, p121) believe that single cases

cannot be of value unless they are linked to hypotheses, which can then be tested systematically with a larger number of cases. Similarly Campbell and Stanley (1966) state that single case studies have a complete absence of control as to be of no scientific value and any appearance of knowledge gained about singular isolated objects is flawed upon closer analysis.

However, as Flyvbjerg (2003) points out, the single case study is well suited to generalising because case studies are centred around falsification, where if just one observation does not conform to the stated proposition then the proposition must either be rejected or revised. Flyvbjerg (2003) goes on to state that falsification is one of the most rigorous tests to which a scientific proposition can be subjected. If propositions are found to be false, it stimulates further investigations and theory building.

Case study research has also been criticised for a lack of rigour in that there is a tendency to reconfirm a researcher's preconceived bias. Diamond (1996, p6) holds this view and observes that the case study does not apply scientific methods, which has the effect of limiting researcher bias. However case study research has no greater researcher bias than any other form of research. In fact as previously stated there is a greater bias towards falsification of propositions than verification (Flyvbjerg, 2003). The lack of rigour in case study research can possibly be attributed to a shortage of texts that provide researchers with specific procedures to be followed (Yin, 2003, p 10).

To ensure a more rigorous approach to the case study research, Yin's (2003, p21) five guidelines will be used:

1. a study's questions;
2. its propositions, if any;
3. its unit(s) of analysis;
4. the logic linking the data to the propositions; and
5. the criteria for interpreting the findings.

The study's research question has already been formulated and according to Yin (2003, p22), "*the case study strategy is most likely to be appropriate for how and why questions*". With this in mind the research question for this study looks at "how" the extended BRD method compares to the RMM method and as such clarifies the nature of the study.

The study's propositions have also been clearly identified in terms of what is to be examined within the scope of the study. As previously stated a comparison of the main concepts, the phases of steps, as well as the modelling technique, notation and graphical representation will be made between the two methods.

The main units of analysis are the two methods under investigation i.e. the extended BRD method and the RMM method. The identification of the units of analysis was made easier by clearly specifying the research question that Yin (2003, p24) points out is fundamental in selecting the appropriate units of analysis. The linking of the data to the propositions is a more difficult task but the general approach is that outlined by Yin (2003, p111), which is to follow the theoretical propositions that led to the case study. In other words the propositions shaped the objectives and design of the case study, which led to the research questions, literature reviews, and ultimately new propositions (Yin, 2003, p112).

The criteria for interpreting the study's findings revolve around the outcomes of applying both methods to the case study used in this research. Campbell (1975) used the approach of pattern matching where several pieces of information from the same case may be related to some theoretical proposition. By using this approach it is hoped that the patterns identified are sufficiently different to either prove or disprove this study's propositions.



## **4. Case Study**

### **4.1 Department of Energy and Technology**

The Department of Energy and Technology (DET) is a medium sized federal government agency based in Canberra. The real identity of the department has been sufficiently disguised to prevent any recognition outside of this study. DET currently employs a staff of about 1200 people across a range of five divisions: Infrastructure, Applications Development, Science, Parliamentary Communications and People Management. DET has been operating for the last 30 years under different guises but maintaining its core function of serving the elected government in developing and implementing energy and technology policy. The applications development team contains 100 analyst programmers who work in small client-focussed teams to provide custom software solutions for the federal government.

In May 2004, DET was asked to provide an online financial information and human resources system that DET staff would access via the Internet. The Web-based information system will allow DET staff to access and change personal details and bank and financial information such as pay details and pay history. In addition, staff will also be able to access and submit leave allowances, overtime claims, training opportunities and lodge travel claims. The main aim of the development of such a system is to remove the manual paper processing of leave, overtime, training and travel claims and allow staff to access personal and financial information quickly and efficiently over the Internet.

### **4.2 The Web-based System**

The Internet based financial and human resources system will consist of five functional areas:

- Personal information.
- Leave information.
- Overtime information.
- Learning and development.
- Travel.

The system is to be implemented using C# in a .Net environment and has tentatively been called Salary and Personal Information (SAPI). The backend database that will be utilised will be SQL server. An implementation using the WWW will provide a wider scope than a traditional client/server implementation as DET has regional offices Australia-wide, some in very remote indigenous areas, and the system will be built to cater for this geographic spread. COTS products were considered but given the resources DET has, particularly in the applications development area, it was more cost effective to custom build a system that would meet the requirements of DET.

### 4.3 Development Strategy

DET Systems Development Methodology (SDM) uses a combination of two types of development processes:

- **Waterfall** – Milestones or gates, i.e. tasks have to be completed before moving onto the next milestone.
- **Spiral** – Continuous assessment of requirements, i.e. making refinements to the project based on feedback.

Each project will have four distinct phases and major milestones as detailed in table 4.1:

**Table 4.1 – Milestones and Phases of DET Systems Development Methodology**

<b>Phase of Project</b>	<b>Milestone</b>
Envisioning	Vision Approved
Planning	Project Plan Approved
Developing	Scope Complete
Stabilising	Completion

The DET SDM Framework is loosely based on four models of the Microsoft Solutions Framework (MSF) and incorporates the following components:

**Process Model** – Improves project control, minimises risk, improves quality, shortens delivery time and provides project structure (project phasing, key milestones, activity identification, deliverables and team relationships and communications).

**Risk Management Model** - Prioritising, decision-making, contingency planning, continuous assessment, and actioning.

**Design Process Model** – Business needs and logical flow; from concept to logical design to physical design, using review and evaluation controls from users, project team and developers.

**Application Model** – Risk driven scheduling, fixed release dates, versioned releases, visible milestones, small peer-based teams (specialist skills for examining outputs/deliverables worked on), identification of critical tasks, planning assumptions and identifying key interdependencies.

## 5. Data Collection and Case Study Preparation

There are numerous case study data collection techniques such as interviews, observation, questionnaires and document and text analysis. Table 5.1 taken from Tellis (1997) describes the strengths and weaknesses of various sources of evidence.

**Table 5.1 – Strengths and Weaknesses of Various Evidence Collection Techniques, taken from Tellis, 1997**

<b>Documentation</b>	
<i>Strengths</i>	<i>Weaknesses</i>
Stable – can be reviewed repeatedly. Unobtrusive – not created as a result of the case study. Exact – contains exact names, references and details of an event. Broad coverage – long span of time, many events and many settings.	Retrievability can be low. Biased selectivity – if collection is incomplete. Reporting bias – reflects (unknown) bias of author(s). Access – may be deliberately blocked.
<b>Archival records</b>	
<i>Strengths</i>	<i>Weaknesses</i>
Same as above for documentation. Precise and quantitative.	Same as above for documentation. Accessibility may be restricted due to privacy reasons.
<b>Interviews</b>	
<i>Strengths</i>	<i>Weaknesses</i>
Targeted – focuses directly on case study topic. May provide causal inferences.	Possible bias due to poorly constructed questions. Response bias. Inaccuracies due to poor recall. Interviewee may provide what interviewer ‘wants to hear’.
<b>Direct observations</b>	
<i>Strengths</i>	<i>Weaknesses</i>
Reality-based – covers events in real time. Contextual – covers context of the event.	Time-consuming. Selectivity – unless there is a broad coverage. Reflexivity – event may proceed differently because it is being observed. Cost – hours needed by human observers

<b>Participant observation</b>	
<i>Strengths</i>	<i>Weaknesses</i>
Same as above for direct observation May provide insights into interpersonal behaviour and motives.	Same as above for direct observation Bias due to investigator's manipulation of events.
<b>Physical artifacts</b>	
<i>Strengths</i>	<i>Weaknesses</i>
May provide insights into cultural features or into technical operations.	Selectivity. May not be available to researcher.

In applying the extended BRD and the RMM to the DET case, the two primary methods that will be used are direct observation and participant observation as the researcher will be applying the methods and documenting the results. In preparation for the case study it was necessary to run through a series of PowerPoint tutorials designed to educate users on how to apply the extended BRD method. Johnstone (2004) who developed the extended BRD method designed the tutorials which have been trialled successfully in a number of action research studies.

It was also necessary to undertake some training on the use of the RMM, however there was very little in the way of tutorials or training materials. Balasubramanian et al. (1996) present a practical application of the RMM to the development of a Web-based system and this example was used as a guideline in applying the RMM to the DET case.

## **6. The Extended Business Rules Diagram Method in Detail**

The Business Rules Diagram (BRD) method was a requirements analysis method developed by McDermid (1998) to assist in the specification of IS requirements. The following are the steps performed when applying the BRD to hypermedia development:

- identify candidate business rules;
- identify candidate events and signals;
- identify candidate objects;
- construct Object Life Histories (OLHs);
- construct User Business Rule Diagrams (UBRDs);
- construct Business Rules Diagrams; and
- construct Event Specification Tables ESTs.

The BRD method was extended by Johnstone (2004) and the following are the additional steps required to be performed.

- construct Process Design Diagrams (PDDs);
- refine Event Specification Tables (ESTs);
- construct Navigation Design Diagrams (NDDs); and
- construct Navigation Tables (NTs).

The extended BRD method is described in detail in the following sections.

### **6.1 Identify Candidate Business Rules**

Candidate business rules are generated using a brainstorming process to develop a possible list of rules, which are then reviewed to refine the list and remove rules that are not relevant to the system (McDermid, 1998). Three types of rules have been identified i.e. policy, processing and implementation rules. Johnstone (2004) defines the rules as follows:

- *Policy Rule: Rules that determine which processes are followed or the way in which objects are treated.*
- *Processing Rules: Rules that describe how processes are followed.*
- *Implementation Rules: The lowest level of business rule often describing the physical implementation of IS.*

The final list of rules are categorised according to which type of rule they represent (Johnstone, 2004).

## 6.2 Identify Candidate Events and Signals

A list of candidate events and signals is obtained by going through a brainstorming process. As McDermid (1998) points out, it is not important to differentiate between events, triggers or messages in the initial stages as it is to compile a list which can then be categorised and refined. Triggers are coded with a T and are those signals that enter a system and initiate activity (Johnstone, 2004). Events are coded with an E and impact on how system components are dealt with internally by an organization (McDermid, 1998). Messages are coded with an M and are used to inform users about events that have occurred inside a system (Johnstone, 2004).

Table 6.1 shows an example of each type. A user receiving the logon prompt is the trigger for the logon process. The user logging on is considered to be an event and once the user has logged on they receive a message confirming that the logon process was successful.

**Table 6.1 – Example of Candidate Events and Signals**

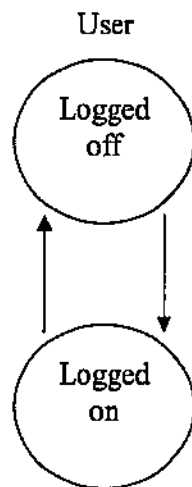
Candidate	Type
User receives logon prompt	Trigger
User logs on	Event
User receives logon confirmation	Message

### 6.3 Identify Candidate Objects

Candidate objects are identified by analysing the lists of events, signals and rules to identify the most relevant objects in the system (Johnstone, 2004). In the example in table 6.1, the only relevant object in the logon process is the user.

### 6.4 Construct Object Life Histories

Each object identified in the system will occupy a number of states over time. The Object Life Histories (OLHs) are constructed by identifying the states for each object in the system and modelling these states (McDermid, 1998). The states an object can occupy are shown as circles and movement from one state to another are shown as arrows (McDermid, 1998). The Harel Blob, as described by McDermid (1998) can also be used to depict a selection of states or states that exist in parallel. An example of an OLH is shown below in figure 6.1.



**Figure 6.1 – Example of an Object Life History (OLH)**







In the example in figure 6.1, a user can move from a state of being logged off to a state of being logged on and back again. The user can occupy either of these states at any point in time.

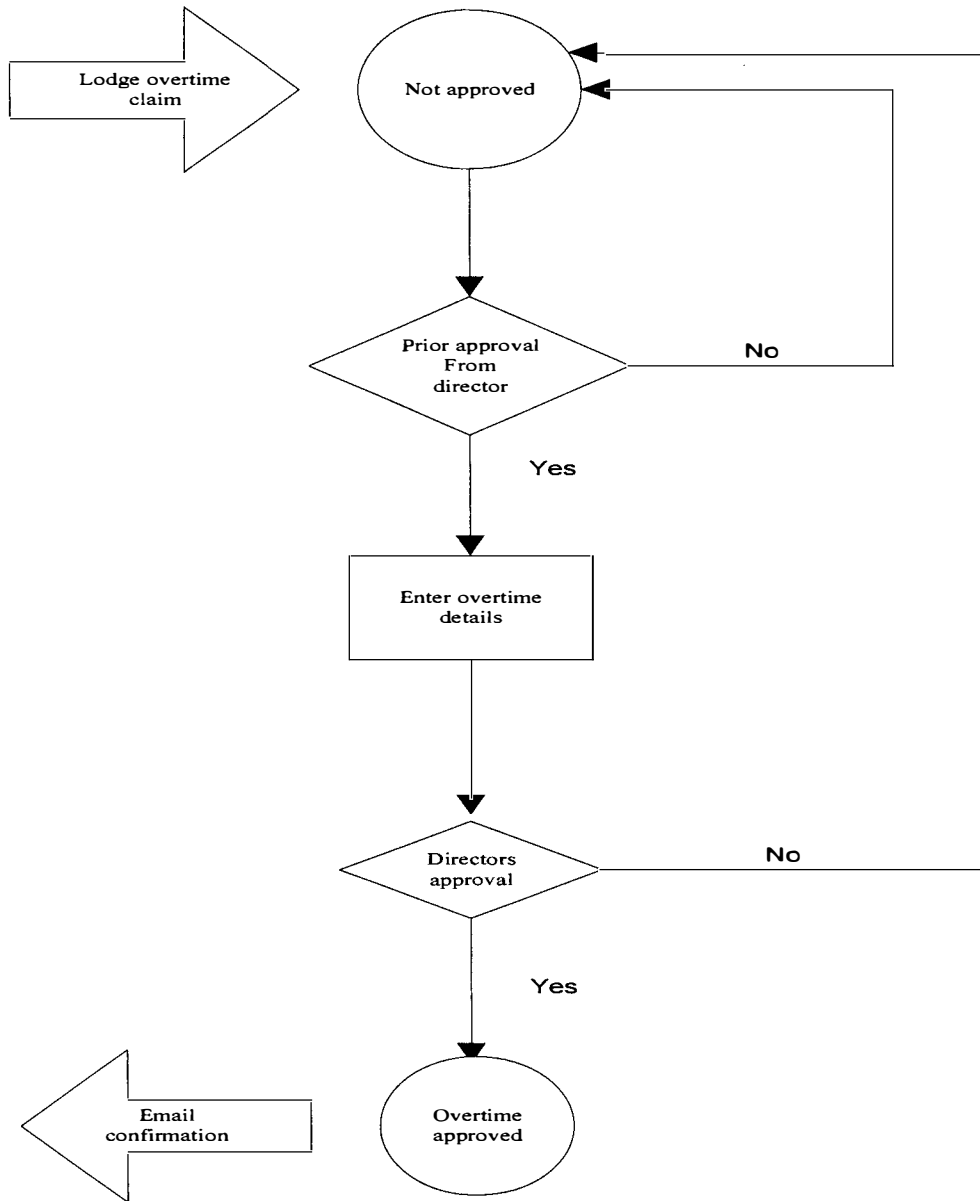


## 6.5 Construct User Business Rules Diagram

Use cases are identified by analysing the OLHs of a particular system and a User Business Rules Diagram (UBRD) is drawn for each use case (McDermid, 1998). By adding the events, triggers and messages identified in step 2 to the OLHs, a UBRD is constructed using the notational constructs detailed in table 6.2 and is used to clarify the business rules concerning the use case. McDermid (1998) goes on to state that it is therefore essential that the diagrammatic notation should be kept as simple as possible to ensure that users find the diagram intuitive. Figure 6.2 shows an example of a UBRD.

**Table 6.2 – Notational Constructs used in the UBRD, taken from Johnstone (2004)**

Symbol	Description
	State. A state reflects the status of an object of interest at any given time
	Event. An event is an action carried out internally by the organisation
	Condition. A condition defines a criterion by which an object of interest in the business moves from one state to another as events take place
	Signal. A signal may either enter or leave the human activity system. A signal that enters the system will typically initiate activity within the system and is called a trigger. A signal which leaves the system serves to inform those outside the system boundary about what has occurred inside the system and is called a message
	Harel blob. The blob encapsulates other constructs and is used to model selection or simultaneous action
	Tag used to connect to one diagram to another. Matching numbers link together.
+	Repetition symbol. The 'plus' superscript is used to indicate one to many instances for states or events in an EST.
,	logical <i>and</i> . The comma represents conjunction and is used in the EST for events that must occur together and in sequences of conditions
<b>bold text</b>	logical <i>not</i> .. Bold text indicates negation and is used in the EST to denote the opposite of a condition



**Figure 6.2 – Example of a UBRD developed in the DET case**

The UBRD shown above looks at the process of lodging and overtime claim. The overtime claim initially occupies a state of “not approved” and moves through to a state of “overtime approved” only if certain conditions have been met. Once the director has approved the overtime claim and the correct overtime details have been entered, an email message is sent to the user confirming approval.

## 6.6 Construct Business Rules Diagram

The step of combining one or more UBRDs into a single diagram is used to construct a Business Rules Diagram (BRD). McDermid (1998) and Johnstone (2004) state that the construction of the BRD is a crucial step and may require refinement through an iterative process to obtain a correct representation of the system. McDermid (1998) and Johnstone (2004) explain that the UBRD differs from the BRD in structure and that each construct in the BRD is uniquely identified using a serial numbering scheme e.g. S1 for state 1, E1 for event 2 and so on. Johnstone (2004) also points out that the states in the BRD must be separated by one or more events i.e. state-event-state. An example of a BRD developed in the DET case study is shown below in figure 6.3.

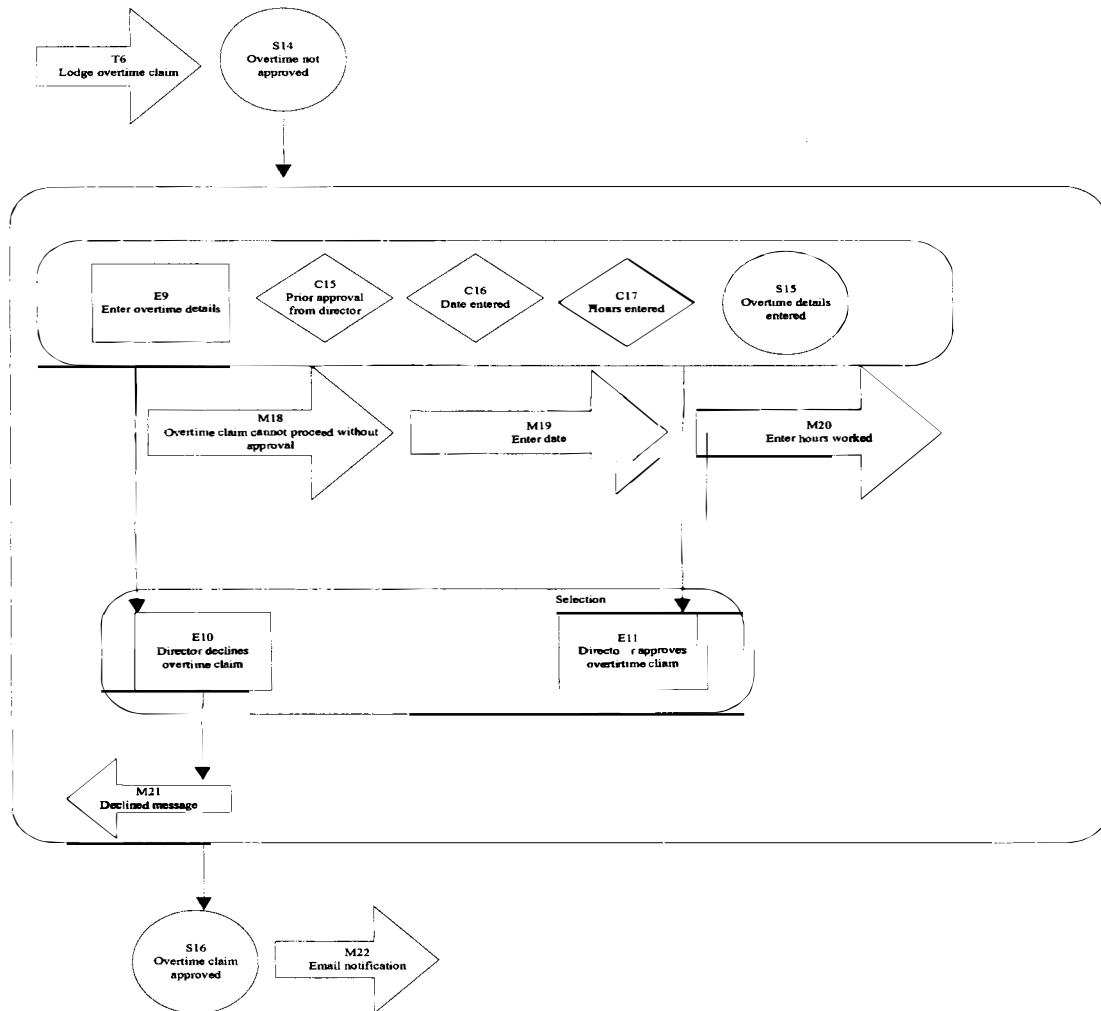


Figure 6.3 – Example of a BRD developed in the DET case.

## 6.7 Construct Event Specification Table

The Event Specification Table (EST) is a way of storing the logic of the BRD by showing the events, triggers, messages, conditions and the state transitions associated with each BRD (Johnstone, 2003). Table 6.3 shows an example of an EST.

**Table 6.3 – Example of an Event Specification Table (EST)**

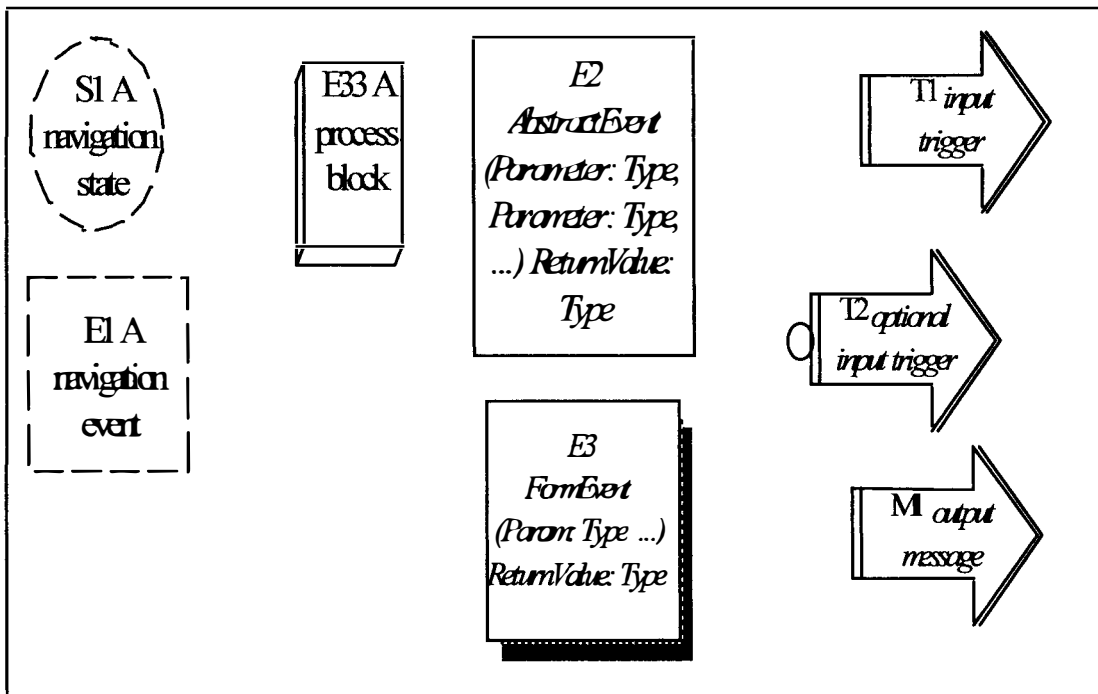
Event	Trigger	Message	Condition	Pre-State	Post-State	Rule
E1	T1	M5	C1 & C2 & C3 & C4	S1	S2, S3	BR1
E1	T1	M[1..4]	!C[1..4]	S1	S1	BR2

In the example in table 6.3, event one (E1) has one trigger, T1 and to move from state one (S1) to state two and three (S2 and S3) conditions one, two, three and four (C1, C2, C3 and C4) need to be true. Once the state transition has occurred message five (M5) is displayed to the user. In the event that conditions one to four are not true (shown as !C[1..4]), E1 does not trigger a state transition and messages one to four (shown as M[1..4]) are displayed to the user.

## 6.8 Construct Process Design Diagram

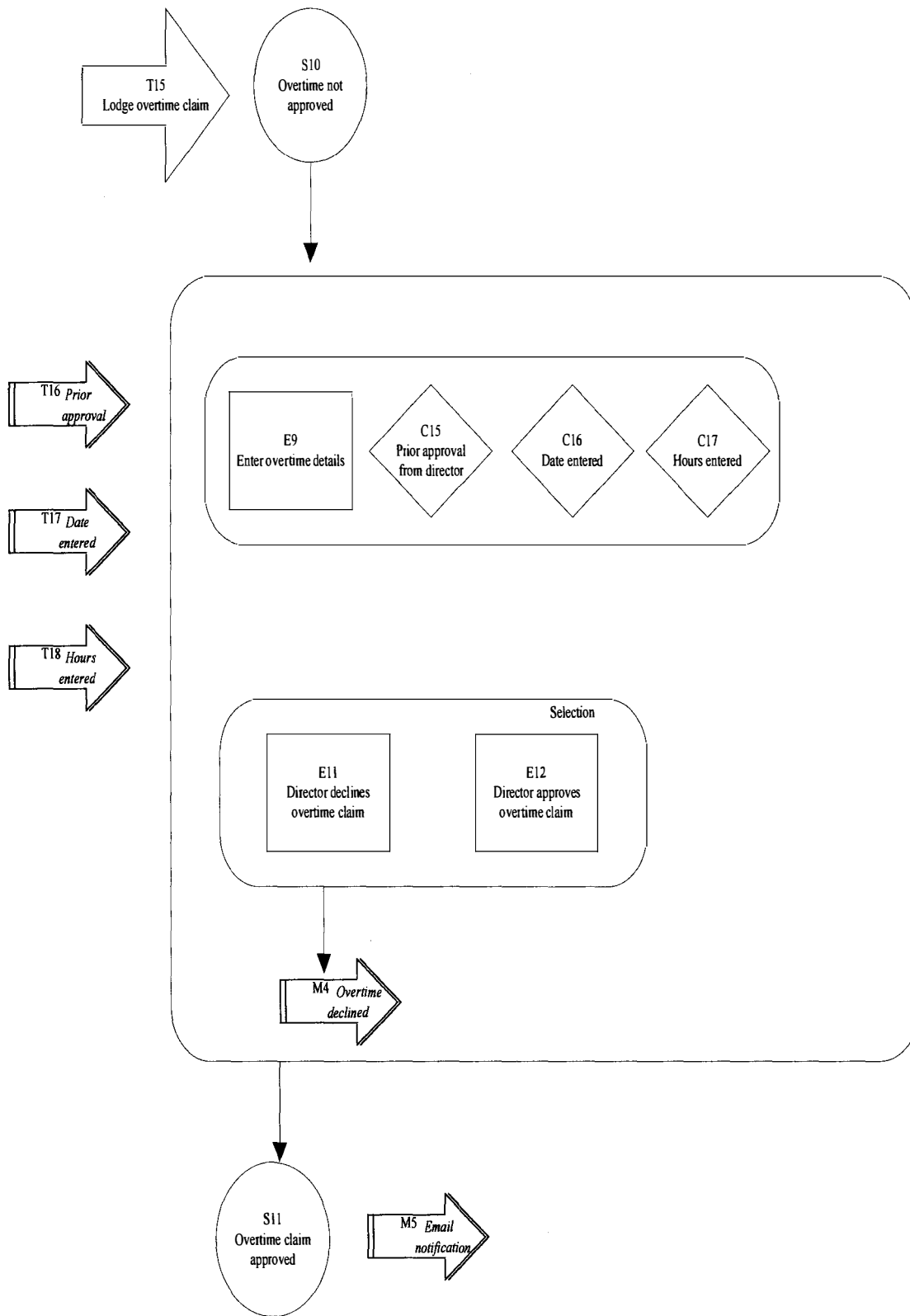
The Process Design Diagram (PDD) is similar to the BRD but uses extended/additional constructs (Johnstone, 2004). The navigation logic is separated from the processing logic and results in separate diagrams for the processing and navigation. The notational constructs used in the PDD have been designed to mirror the symbols already used in the BRD. Johnstone (2004) also deliberately chose symbols that were not technology specific, as the technologies used in Web-based systems tend not to be stable. Figure 6.4 shows the notational constructs used in the PDD and Johnstone (2004) describes the design symbols as follows:

“processing-oriented events are shown with the same rectangle as events in the BRD. An abstract event is one for which the technology is not (yet) chosen. If a technology is chosen, then the technology is given in the name and the rectangle is shown with a shadow. Similarly, the process block includes the choice of technology, if known (e.g. API, servlet, applet, script etc.), as text within its symbol”.



**Figure 6.4 – Extended Notation for the Process Design Diagram, taken from Johnstone et al. (2003)**

Figure 6.5 overleaf shows an example of a PDD.



**Figure 6.5 – Example of a PDD for Lodge Overtime Claim developed in the DET Case**

## 6.9 Refine Event Specification Tables

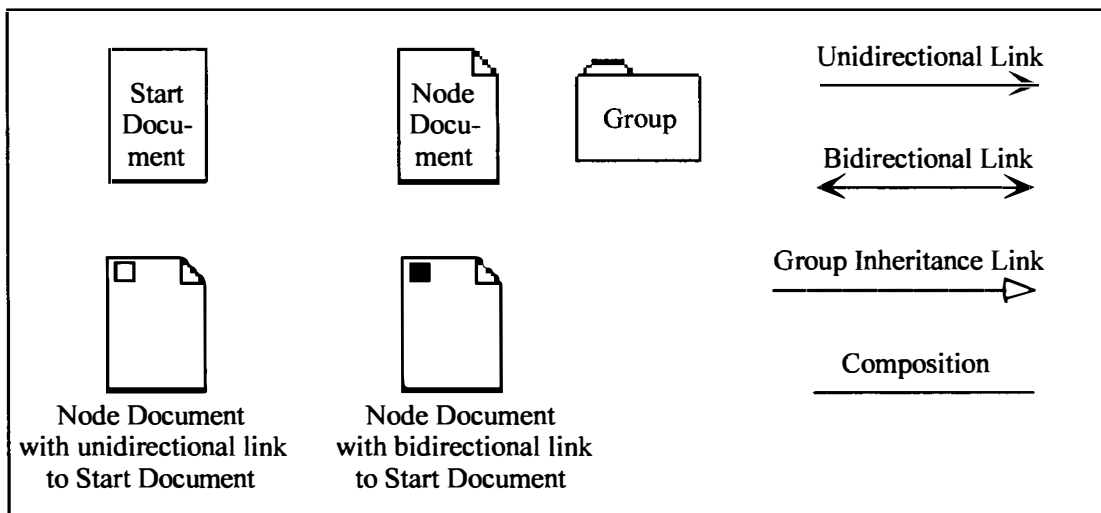
The same way in which a BRD has an EST associated with it to confirm the business logic, a PDD has a refined EST associated with it. An example of the EST associated with the PDD from the Lodge Overtime Claim example in table 6.4 is shown below.

**Table 6.4 – Refined EST for Overtime PDD developed in the DET case**

Event	Trigger	Message	Condition	Pre-State	Post-State	Rule
E9	T15, T16, T17, T18	M5	C15, C16, C17	S10	S11	3.1, 3.2, 3.3
E11	T15, T16, T17, T18	M4	!C[15..17]	S10	S10	3.1, 3.2, 3.3
E12	T15, T16, T17, T18	M5	C15, C16, C17	S10	S10	3.1, 3.2, 3.3

## 6.10 Construct Navigation Design Diagrams

The Navigation Design Diagram is used to model the navigation paths that a user will traverse in the final system. The notational constructs used in the NDD are detailed in figure 6.6 below.



**Figure 6.6 – Notation used in the Navigation Design Diagram, taken from Johnstone et al. (2003)**

The full NDD for SAPI is detailed in figure 8.6.

### 6.11 Construct Navigation Tables

The final step in the extended BRD method is to construct a navigation table (NT). The NT consists of a matrix in which the site links and destination links identified in the NDD are mapped in a matrix. The links are represented as either uni or bi-directional. In the example below in figure 6.7, which is duplicated later in the DET case study, Personal Details, Leave, Overtime and Learning and Development all have bi-directional links to the home page. While this only represents a partial navigation table it depicts the basic structure. Matrices vary in size and more complex systems will lend themselves to larger matrices. Johnstone (2004) suggests the use of embedded XML links in the NT and the complexity will be reflected in the way these links are used.

Site Link Dest Link	Home Page	Personal Details	Leave	Overtime	Learning and Development (L&D)
Home Page		<-->	<-->	<-->	<-->
Personal Details	<-->				
Leave	<-->				
Overtime	<-->				
Learning and Development (L&D)	<-->				

**Figure 6.7 – Partial Navigation Table developed in the DET case**



## 7. The Relationship Management Methodology (RMM) in Detail

Isakowitz et al. (1995) developed the Relationship Management Methodology (RMM) as a vehicle for managing relationships among information objects. The applications for which RMM is best suited exhibit a regular structure that contains classes, relationships between classes and multiple instances of objects for each class (Isakowitz et al., 1995). The data that is contained in these applications tends to be volatile and the RMM provides a framework for the design of such applications. Isakowitz et al. (1995) describe seven steps that make up the RMM some of which can be applied concurrently. The following is a description of each step in detail.

### 7.1 The Entity Relationship Diagram

The first step is the Entity Relationship (E-R) Design, which looks at the relevant entities and the relationships between those entities of the application domain. These entities and relationships will end up as nodes and links in the resulting hypermedia application (Isakowitz et al., 1995). During this step it is important to think about the entities from the point of view of their interrelationships and the information a user will want to access (Balasubramanian et al., 1996). Figure 7.1 shows a simple example of an E-R diagram that models the entities and relationships that might occur in a banking environment.

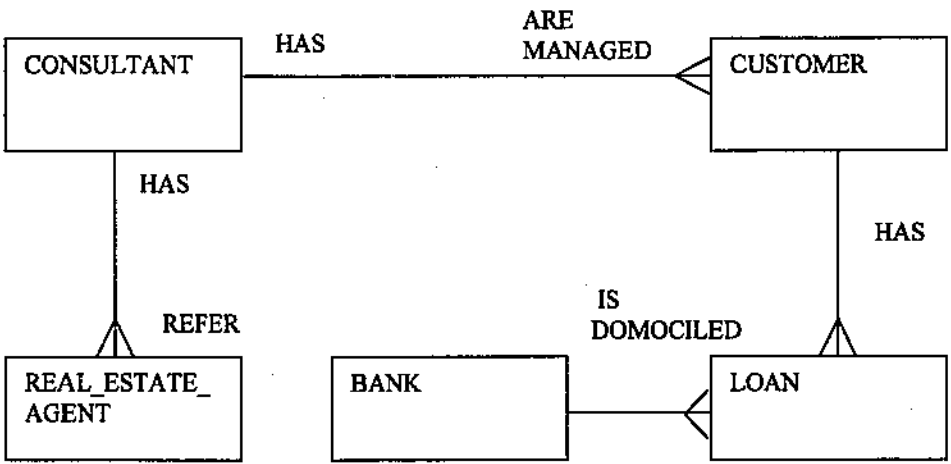


Figure 7.1 – Entity Relationship Diagram

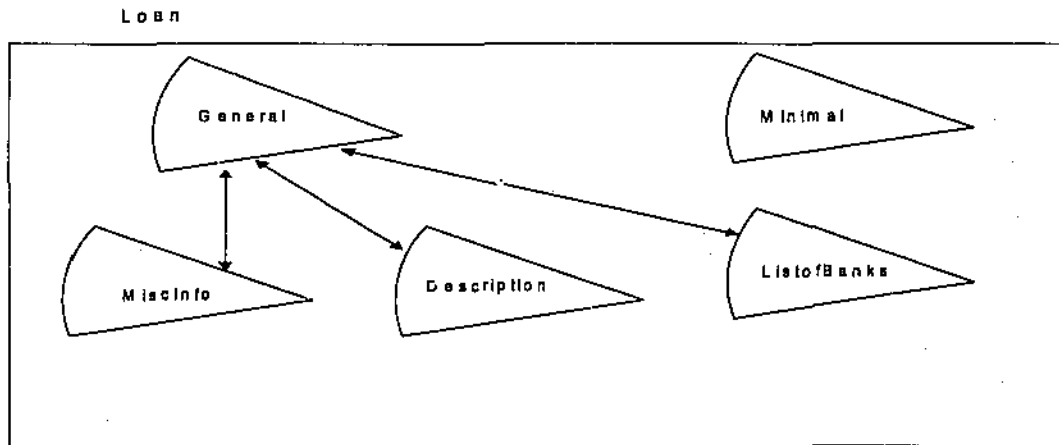
The relationships that are shown in the E-R diagram are represented as either one to one or one to many. For example, in figure 7.1 the consultant entity has a one to many relationship with the customer entity i.e. a consultant has many customers. The main objective in the E-R design step is to ensure that the important relationships are identified and that the links between the objects are explicit as the links represent the main paths an end user will traverse to navigate between objects (Balasubramanian et al., 1996).

## **7.2 Slice Design**

Each entity identified in the E-R diagram has a number of attributes and the second step in RMM, the slice design, involves grouping these attributes for display. As Isakowitz et al. (1995) point out, an entities information can be displayed in one window using scroll bars but this approach has its drawbacks. The impact of the information may not be as great if users have to constantly scroll, particularly within large windows. The Slice Design step addresses this by attempting to group together meaningful units of information that can be displayed on one page yet still remain interrelated.

Entities are split into meaningful slices that are organised into a hypertext network. Each slice groups one or more attributes of an entity and each minimal slice (M-Slice) groups attributes of collections of entities. M-Slices model what information is to be part of a presentation unit, not how this information will actually be presented. The default target of access structures entering an entity is called the head slice, which is connected to all the other slices via structural links that can be uni or bi directional.

For example, the loan entity from figure 7.1 has the following attributes; LoanName, LoanType, LoanNumber, Date, Description, ListofBanks and MiscInfo. This can be split into four regular slices i.e. General, Description, ListofBanks and MiscInfo, which is shown in figure 7.2 overleaf.



**Figure 7.2 – Slice Diagram**

The attributes for each slice are grouped as follows:

**General:** LoanName, LoanType, LoanNumber, Date

**Minimal:** LoanName, LoanType

**Description:** LoanName, Description

**ListofBanks:** LoanName, ListofBanks

**MiscInfo:** LoanName, MiscInfo

The general slice in this instance serves as the head slice and the minimal slice (M-Slice) contains the minimum set of context information necessary to identify this entity. The links between slices in the loan slice diagram are called structural links. These are links that join information pieces within the same entity instance (Isakowitz et al., 1995). Relationships between different entity instances that belong to different entity classes are described as associative relationships and are represented as dashed lines in the final RMD diagram (e.g. the link between Bank and Loan in figure 7.1 is an associative relationship). The end result is what Isakowitz et al. (1995) describe as an enriched E-R diagram, denoted as E-R<sup>+</sup>.

There are no hard and fast rules when attempting the slice design but Isakowitz et al. (1995) outline four main considerations:

1. dividing an entity into slices;
2. choosing one slice to be the head of the entity;
3. interconnecting the various slices; and

#### 4. labelling the links.

When dividing an entity into slices it must be remembered that a slice represents a whole to the user, therefore slices should only group related information. The head slice should be the most representative of the slices with the links reflecting the need to connect more specialised slices to more general ones, with the labels becoming the anchors (Isakowitz et al., 1995).

### 7.3 Navigational Design

Step 3 is the Navigational Design and involves designing the paths that will make hypertext navigation possible. Each of the associative and structural relationships that appear in the E-R<sup>+</sup> diagram are analysed to determine if they should be accessible for navigation in the final system. If it is decided they are to be made accessible then they are replaced by one or more RMDM access structures (Balasubramanian et al., 1996). The RMDM access structures are detailed in Figure 7.3 below.

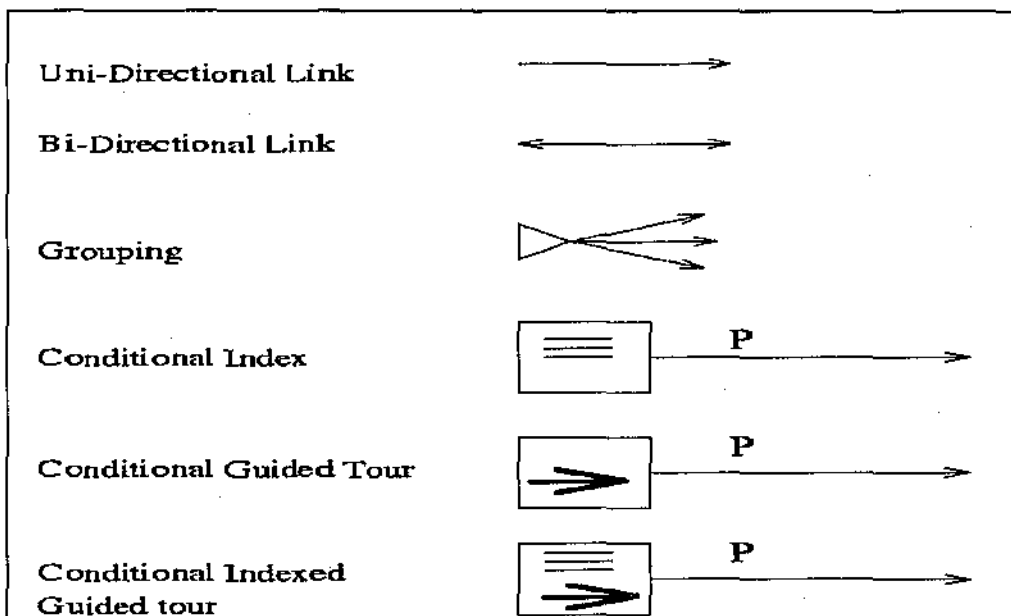
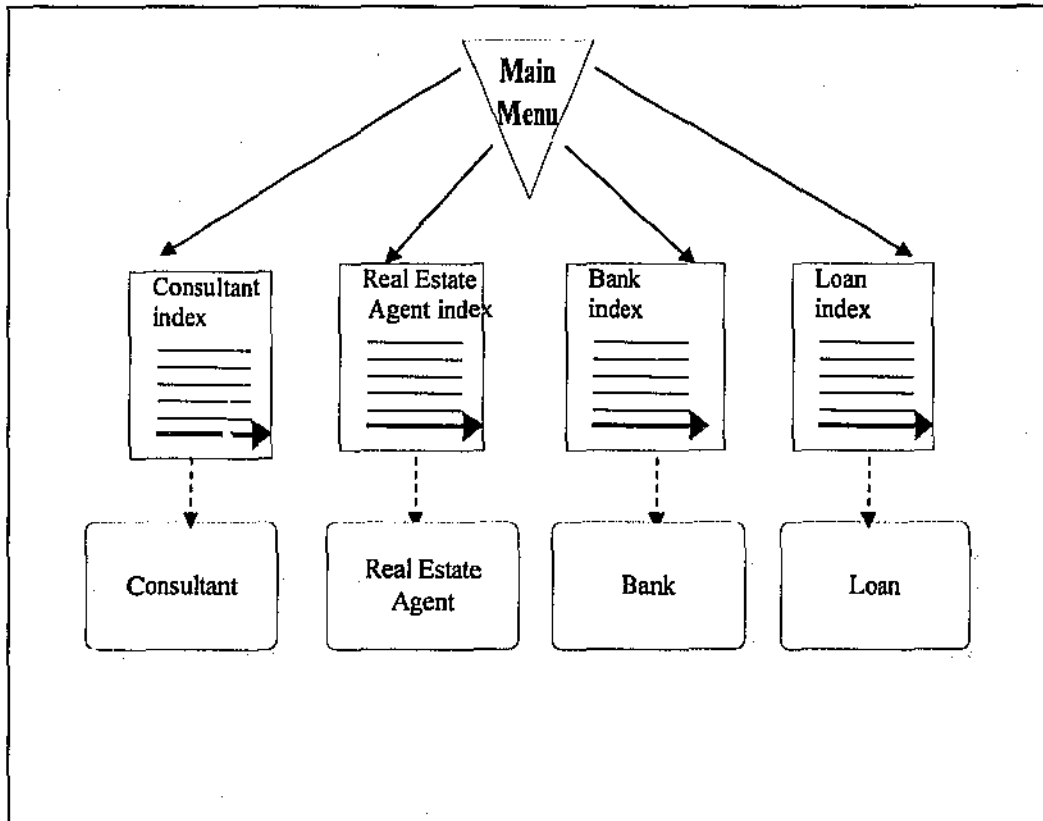


Figure 7.3 – The elements of the RMM Data Model (RMDM), taken from Balasubramanian et al., (1996)

As explained by Balasubramanian et al. (1996), Navigation is achieved via the six primitives detailed in figure 7.3. The uni and bi-directional links specify the types of access between slices of an entity. The grouping mechanism provides a menu like mechanism supporting access to other parts of the application but has largely been replaced by minimal slices (M-Slices). An index is a table of contents to a list of entity instances that provide access to all the elements of the list directly. A Conditional Guided Tour provides a linear path either backwards or forwards through a collection of items. In addition Indices and Guided Tours can also be combined to give Conditional Indexed Guided Tours. The Indices and Guided Tours each have logical predicates specifying which entities participate. For example, a condition “type=artist” attached to an index into a *concert\_event* entity denotes an index to artists from that concert.

Figure 7.4 shows a partial RMDM diagram based on the E-R diagram in figure 7.1.



**Figure 7.4 – Partial RMDM Diagram**

#### **7.4 Conversion Protocol Design**

Isakowitz et al. (1995) provide only brief details for the remaining four steps because they are related more to the physical design of the system and not the design of access mechanisms (Balasubramanian et al., 1996). The fourth step, *conversion protocol design*, produces a set of rules that governs how to transform each element of the RMDM diagram into an appropriate object in the implementation environment (Isakowitz et al., 1995). The logical E-R design is translated into a physical design with the focus very much on a structural transformation.

#### **7.5 User Interface Design**

The fifth step, *user-interface design*, takes each object in the RMDM diagram and considers the actual presentation of each object in the design of suitable interface components (Balasubramanian et al., 1996). This includes the overall look and feel, the look of anchors, button layouts and positioning of multimedia and images.

#### **7.6 Runtime Behaviour Design**

In the sixth step, *runtime behaviour design*, the focus is on the functionality that supports the runtime behaviour, which includes but is not limited to history lists, link traversal, navigational mechanisms and dynamic generation versus static generation of pages (Balasubramanian et al., 1996).

#### **7.7 Construction and Testing**

In the final step the application is actually constructed and tested. Isakowitz et al. (1995) state that the testing should ensure that all navigational paths are operational and Balasubramanian et al. (1996) expand on this by adding that the application should satisfy all functional, navigational and usability requirements after thorough testing.

## 8. The Extended Business Rules Diagram Method Applied to DET

### 8.1 Identification of Candidate Business Rules

It was decided to apply the extended BRD to the DET case first as it would provide a set of user requirements that could be used in the RMM method. During the first brainstorming session only 10 business rules were identified. These were mainly policy rules and the lack of a large list can be attributed to an emphasis being placed on only identifying policy rules and discounting the processing or implementation rules. A second attempt resulted in generating more than 70 business rules that covered more completely the spectrum of policy, processing and implementation categories. Table 8.1 shows an example of the types of rules generated.

**Table 8.1 – Example of candidate business rules**

<b>Rule</b>	<b>Description</b>
1.1	A DET employee can only access SAPI if they are a current permanent employee and have a valid activated DET LAN logon.
1.2	Only a system administrator can activate a DET LAN logon.
1.3	A DET employee can only access their own SAPI information after logging onto the DET network using their DET LAN logon credentials.
1.4	A DET employee can only open one active SAPI session at a time.
1.5	Only human resources staff can access and change/add restricted SAPI staff details, such as salary details and work schedule.
1.6	A DET employee can only lodge a leave claim if they have accrued personal/annual leave.
1.7	A DET employee must provide a medical certificate if more than two days personal leave is taken.
1.8	All leave claims must be authorised by an employee's director.

After the generation of the business rules, the list was culled slightly and the rules were categorised as either policy, processing or implementation rules. This step proved to be quite intuitive, especially since the basic requirements of the SAPI system was known prior to the brainstorming session. It was important not to discount any rule generated but rather capture as many rules as possible and conduct the culling process (if any rules were to be culled) during the categorisation phase.

The categorisation of the rules proved to be a little more difficult however, as in a number of cases determining whether a state(s) was externally verifiable by a user could be argued either way. It was also later identified that a large number of the rules generated fell into the processing category. Rules could also be refined or added as the development process continued.

## 8.2 Identification of Candidate Events and Signals

This step also proved to be quite intuitive, given that Johnstone (2004) identified events as being the verbs in each business rule. Once the events were identified it was a natural progression to identify the triggers for each event and the associated messages that might occur as a result of an event. Table 8.2 shows an example of the events, triggers and messages identified for Use SAPI.

**Table 8.2 – Example of Events, Triggers and Messages for Use SAPI.**

<b>Candidate</b>	<b>Type</b>
User receives logon prompt	Trigger
User logs on	Event
User receives successful logon confirmation	Message



### 8.3 Identification of Candidate Objects

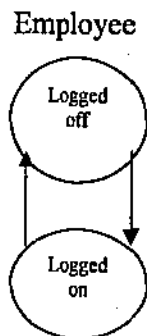
Using Johnstone's (2004) strategy of identifying the nouns in a business rule to identify the objects made the translation of this step easier. Having said this, upon further analysis, a small number of the objects identified did not change state and were later discarded as it was decided they were not relevant to the system. Table 8.3 shows an example of some of the candidate objects that were identified.

**Table 8.3 – Example of Candidate Objects Identified**

Object	Description
Employee	A person that has been engaged as an employee by DET and assigned an AGS number.
LeaveClaim	An application lodged by an employee when requesting personal or annual leave.

### 8.4 Construction of Object Life Histories

This step was in essence completed in parallel with the previous step. Objects were identified and if they did not generate life histories they were discarded. The remaining Object Life Histories (OLHs) were modelled and refined. It was interesting to note that if a candidate business rule was simple, the corresponding OLH was simple as well. As a result some of the candidate business rules were refined after the OLHs were modelled. An OLH for employee is shown in Figure 8.1.



**Figure 8.1 – OLH for Employee**

## 8.5 Construction of User Business Rules Diagrams

It was initially difficult to construct the User Business Rules Diagram (UBRD) from the OLHs, so it was decided to follow a recommendation proposed by Johnstone et al. (2003), which was to model use case diagrams from the business rule set. As a result the construction of the UBRDs became easier as the business logic of the events and signals translated well from the use case diagrams. An example of a use case diagram and UBRD for Use SAPI is shown below in figure 8.2 and 8.3 respectively.

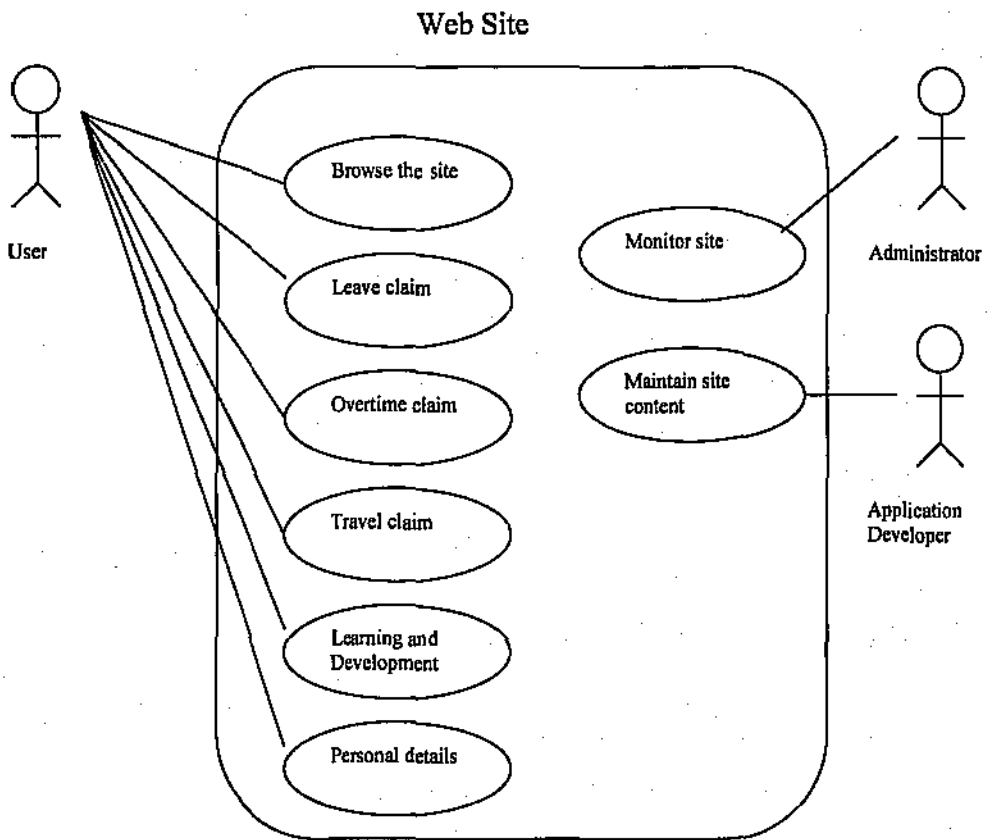
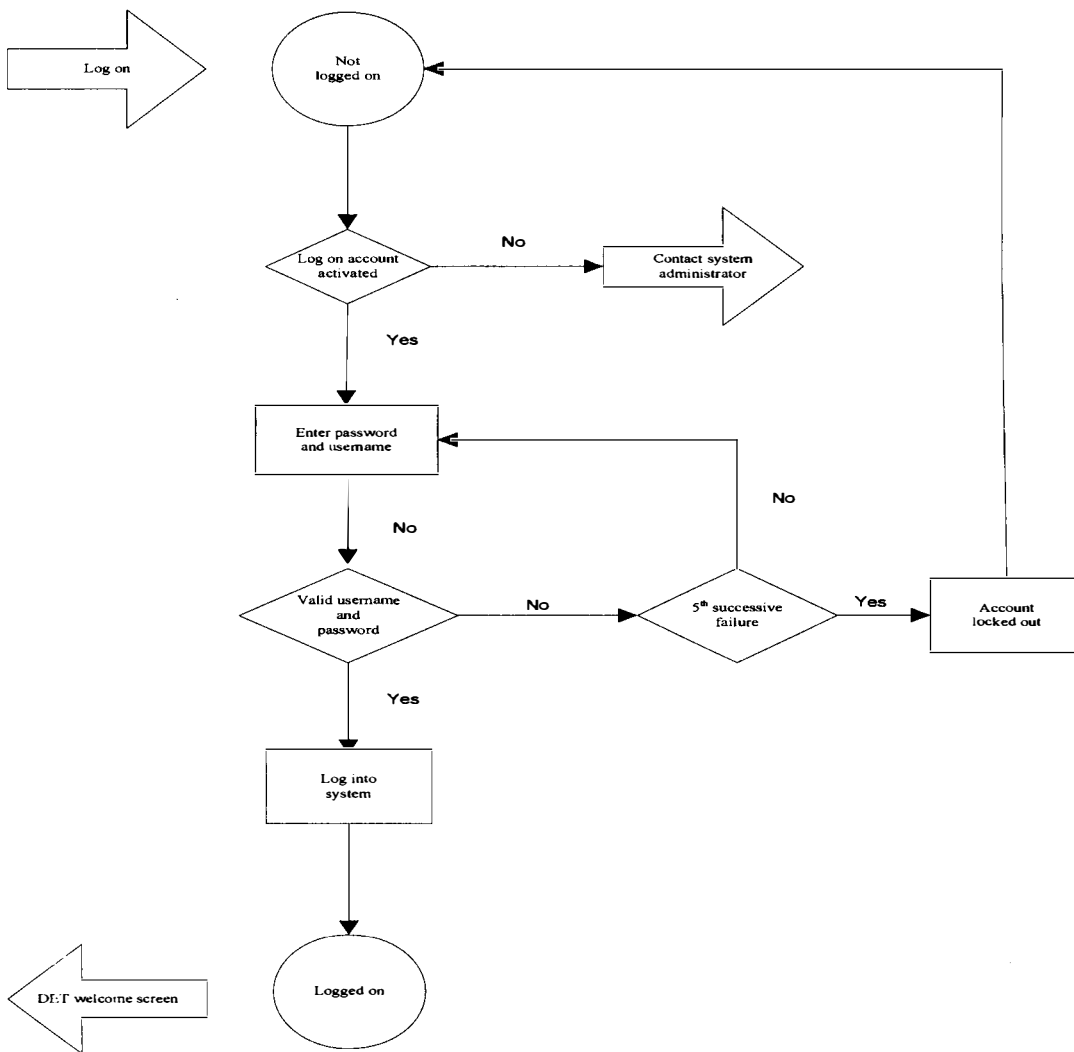


Figure 8.2 – Use Case Diagram for SAPI

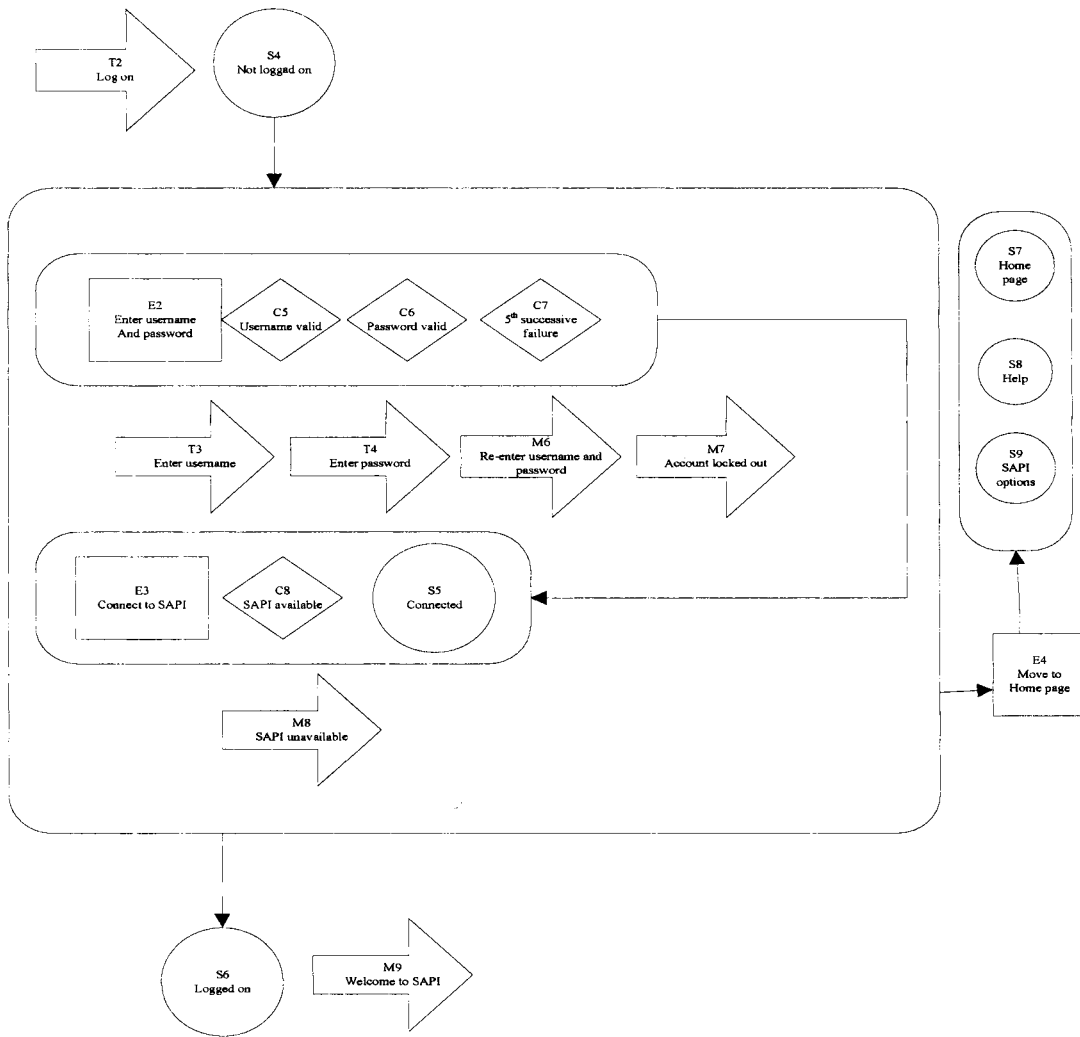


**Figure 8.3 – UBRD for Use SAPI**

## 8.6 Construction of Business Rules Diagram

The construction of the Business Rules Diagram (BRD) was made easier by the fact that the notational constructs used in the modelling of the BRD were already introduced during the construction of the UBRDs. While the structure of the BRD may not resemble the UBRD, the general principle is the same when constructing the BRD. The only differences as Johnstone et al. (2003) point out are that additional states and constructs are added and they are grouped using the Harel blob. This allows for a far more granular diagrammatic representation of the business rules. The BRD was refined several times for each UBRD to ensure that the BRD was valid and

accurately represented the business rules. The BRD for Use SAPI is shown in figure 8.4.



**Figure 8.4 – BRD for Use SAPI**

### 8.7 Construction of Event Specification Tables

The construction of the Event Specification Table (EST) involved taking the events and signals from the BRDs and presenting them in a tabular format. Due to the refinement of the BRDs in the previous step, the ESTs required little modification and were able to be linked to a corresponding business rule, confirming the validity of the BRDs. Each row in the EST satisfied the Meta rule described by Johnstone et al. (2003) i.e. state-event-state.

The Use SAPI EST is shown in table 8.4.

**Table 8.4 – EST for Use SAPI**

<b>Event</b>	<b>Trigger</b>	<b>Message</b>	<b>Condition</b>	<b>Pre-State</b>	<b>Post-State</b>	<b>Rule</b>
E2	T2	M9	C5,C6,!C7	S4	S6	1.1
E2	T2	M6	!C[5..7]	S4	S4	1.1
E2	T2	M7	!C[5..6], C7	S4	S4	1.1, 1.7
E3		M9	C8	S6	S5	1.1, 1.5, 1.6, 1.7
E3		M8	!C8	S6	S6	1.1, 1.5, 1.6, 1.7
E4			C5, C6 ,!C7, C8	S4, S5 S6	S7, S8, S9	1.1, 1.4, 1.5, 1.6, 1.7, 1.8, 1.10

For example, in the first row of table 8.4, a user moves from a state of not being logged on (S4), to a state of being logged on (S6), after receiving the triggers T2, T3 and T4. The conditions that must be satisfied for the state transition to occur are C5, C6 and not C7. Should any of the conditions not be satisfied, as shown in the second row of table 8.4, the state transition does not occur.

## **8.8 Construction of Processing Design Diagrams**

The construction of the Processing Design Diagrams (PDD) required the researcher to familiarise himself with the additional design notations used in the PDD. Once this was done it was relatively easy to identify the navigation components and the processing components and start to develop the PDD by separating them out. The design notation used in the PDD was not that dissimilar to the design notation used in the BRD, which made the transition to the PDD easier. The Use SAPI PDD is shown overleaf in figure 8.5.



**Table 8.5 – Event Specification Table for Use SAPI**

Event	Trigger	Message	Condition	Pre-State	Post-State	Rule
E5			C8	S4	S6	1.5, 1.6, 1.7, 1.8, 1.10
E2	T2, T3, T4, T6			S4	S7, S8, S9	1.1, 1.4
E3	T7		C5	S4	S4	1.1, 1.4
E4	T8		C6, C7	S4	S7, S8, S9	1.1
E4	T8		C6, C7	S4	S4	1.1
E6				S4	S6, S7, S8	1.7, 1.8

**8.10 Construction of Navigation Design Diagrams**

The Navigation Design Diagrams (NDDs) was developed by revisiting the BRDs and the use case diagrams. The researcher again had to familiarise himself with the design notations used in the NDDs before starting the NDD construction. While this step comes after the construction of the PDDs it was found to be of more benefit to construct the NDD before the PDDs. Starting the construction of the NDD prior to the PDD enabled a clearer identification of the processing components and the navigation components that would be separated out in the PDD.

What was also found to be of benefit was encapsulating a number of global links in a Harel blob. While this is not described in the eBRD it was used in one of Johnstone’s (2004) action research studies. The benefit of using the Harel blob is that it enables the linking of certain nodes to all leaf nodes implicitly. What was also interesting was the fact that nearly all the node documents in the DET NDD had the same bi-directional and uni-directional links back to their respective groups.

For example, the “Amend Details” node in figure 8.6 has a uni-directional link to the “Confirm Changes” node. The purpose of the confirm changes node is to take the user to a new page and display all the changes made. If a user is satisfied with the





### 8.11 Construction of the Navigation Table

Construction of the Navigation Table involved duplicating the logic developed in the NDD and representing it in tabular form. The Navigation matrix developed measured 24 x 24 and due to size restrictions only a partial representation is reproduced in figure 8.7 below.

Site Link Dest Link	Home Page	Personal Details	Leave	Overtime	Learning and Development (L&D)
Home Page		<-->	<-->	<-->	<-->
Personal Details	<-->				
Leave	<-->				
Overtime	<-->				
Learning and Development (L&D)	<-->				

**Figure 8.7 – Partial SAPI Navigation Matrix**

## 9. The Relationship Management Methodology Applied to DET

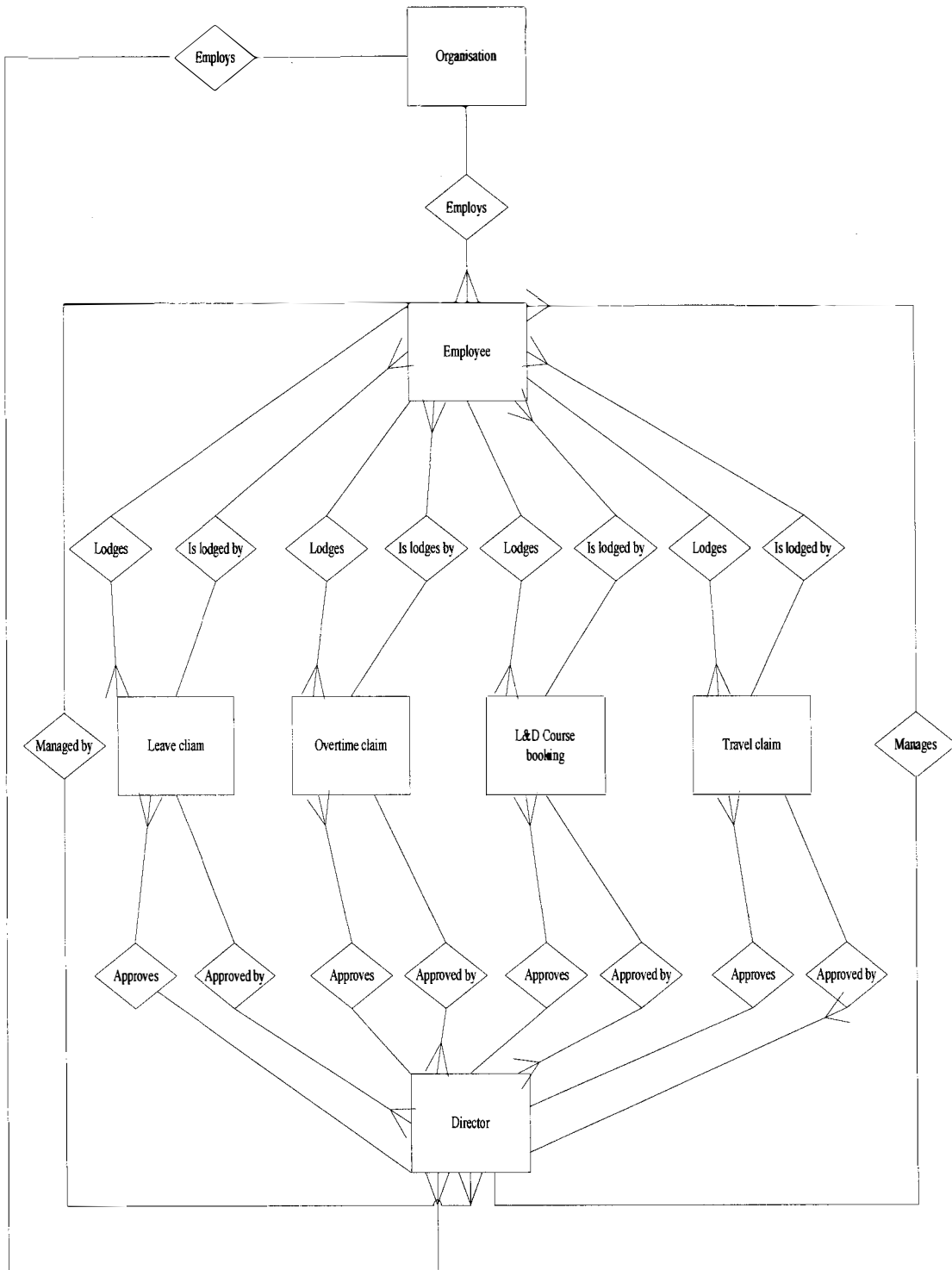
### 9.1 The Entity Relationship Design

As previously stated, the business rules generated as part of the EBRD method were used to identify the user requirements that could then be used as a basis for the system development using the RMM method. The rationale for this approach is justified, as there is no formal step in the RMM method that specifically addresses user requirements. The business rules were analysed to identify the main entities and the relationships that exist between those entities.

Based on the requirements, eight entities were identified and eight associative relationships. Table 9.1 lists the entities and their attributes and figure 9.1 overleaf shows the Entity Relationship (E-R) diagram.

**Table 9.1 – List of entities and their attributes**

<b>ENTITY</b>	<b>ATTRIBUTES</b>
Employee	FirstName, LastName, AGS No, DOB, HomeAddress, WorkAddress, WorkPhone no, HomePhone no
Director	FirstName, LastName, AGS No, DOB, HomeAddress, WorkAddress, WorkPhone no, HomePhone no
Leave Claim	Leaveno, TypeofLeave, AccruedLeave, Dates
Overtime Claim	OvertimeNo, HoursWorked, Dates
L&D Course Booking	Courseno, Type, Name, Date, Duration, Location
Travel Claim	Travelno, Origin, Startdate, Enddate, Destination, MealsClaimed
Bank Account	BSBno, Accountno, Location
Organisation	OrgName, Orgtype, Address, Phone, Fax, Email

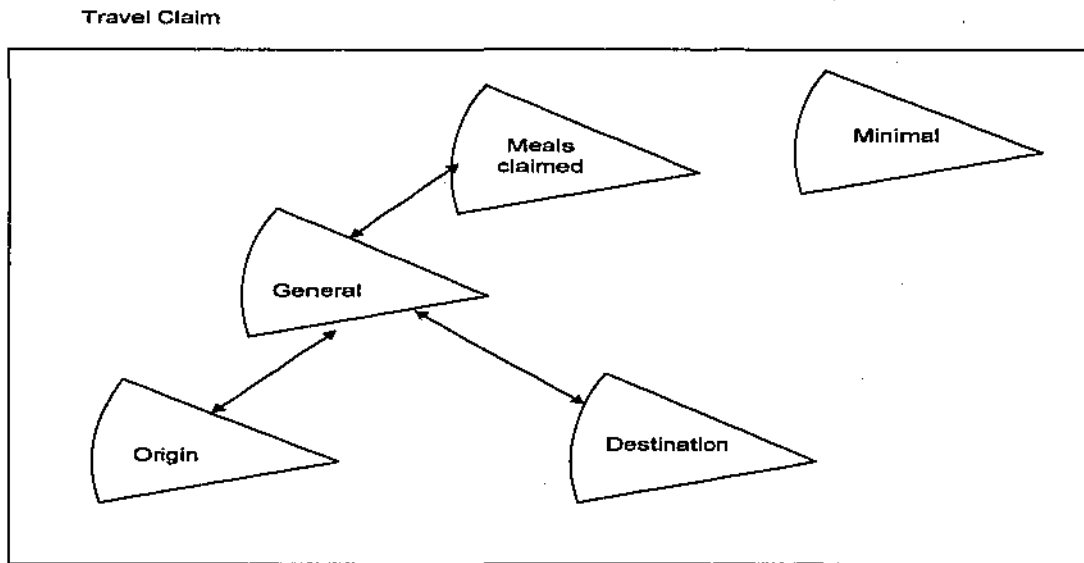


**Figure 9.1 – E-R diagram for SAPI**

## 9.2 Slice Design

During this step each entity was examined to determine if there was a need to slice each entities attributes into subsets. This process was quite difficult because even

though Isakowitz et al. (1995) provide general guidelines, the choice of which attributes should be grouped together or even kept at all depends on the viewpoint of the developer. It was difficult to envisage how much information was appropriate to display in each slice. It was also difficult determining the maximum number of slices that would avoid fragmentation. This step was similar in some ways to the normalisation process found in database design techniques. The slice design for the entity travel claim is shown in figure 9.2 along with the attributes for each slice.



Slice	Attributes
General	Travelno, StartDate, Enddate
Minimal	Travelno
Origin	Travelno, Destination
MealsClaimed	Travelno, MealsClaimed

**Figure 9.2 – Slice Design for Travel Claim and corresponding attributes**

### 9.3 Navigation Design

The Navigation Design step was fairly intuitive. The entities associative and structural relationships were examined to determine if they were valid and also what access structure would be used for that entity in the final system. Isakowitz et al. (1995) The Relationship Management Data Model (RMDM) access structures were

revisited to ensure the relationships were accurately represented. The high-level access structures were then created by grouping items of interest. Figure 9.3 shows the Navigation Design Diagram for the SAPI system.

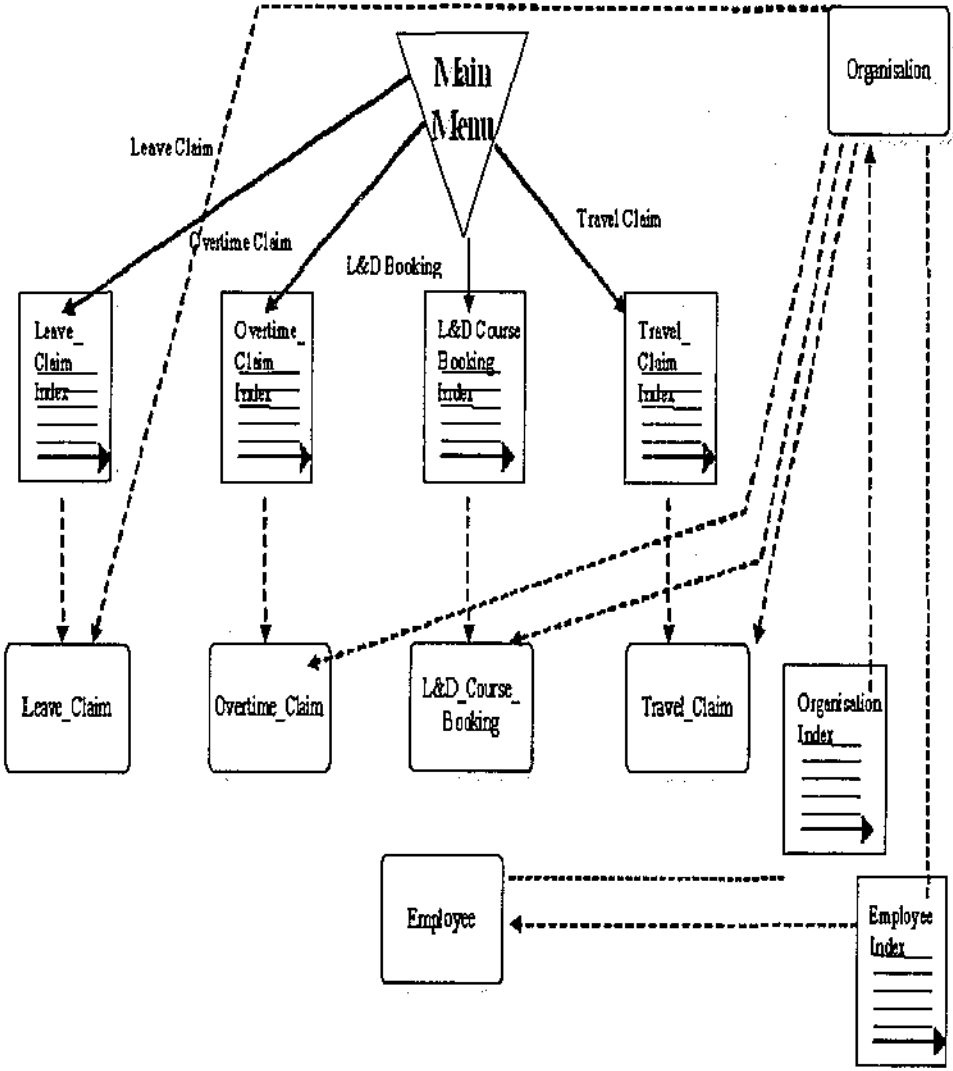


Figure 9.3 – NDD for SAPI

## 9.4 Conversion Protocol Design

Isakowitz et al. (1995) do not describe this step in any detail except to say that “conversion protocol design uses a set of conversion rules to transform each element of the RMDM diagram into an object in the target platform e.g. a toolbook list-box or HTML form can be used to implement an index”. As there are many options and approaches available for protocol conversion, the researcher decided to use HTML forms to implement indexes as it integrates well with Web-based systems.

## 9.5 User-Interface Design

The user-interface design was constructed by examining the NDD and the RMDM diagram. This step is concerned with the screen layouts of the diagrams created in step 3 and allows the developer to be creative with the look and feel of the pages. It was decided to produce some HTML code that would represent the layout of the buttons, nodes and provide a description about how a user would traverse links and backtrack if necessary. A screen shot of the DET home page and the Travel page are shown in figures 9.4 and 9.5 below.

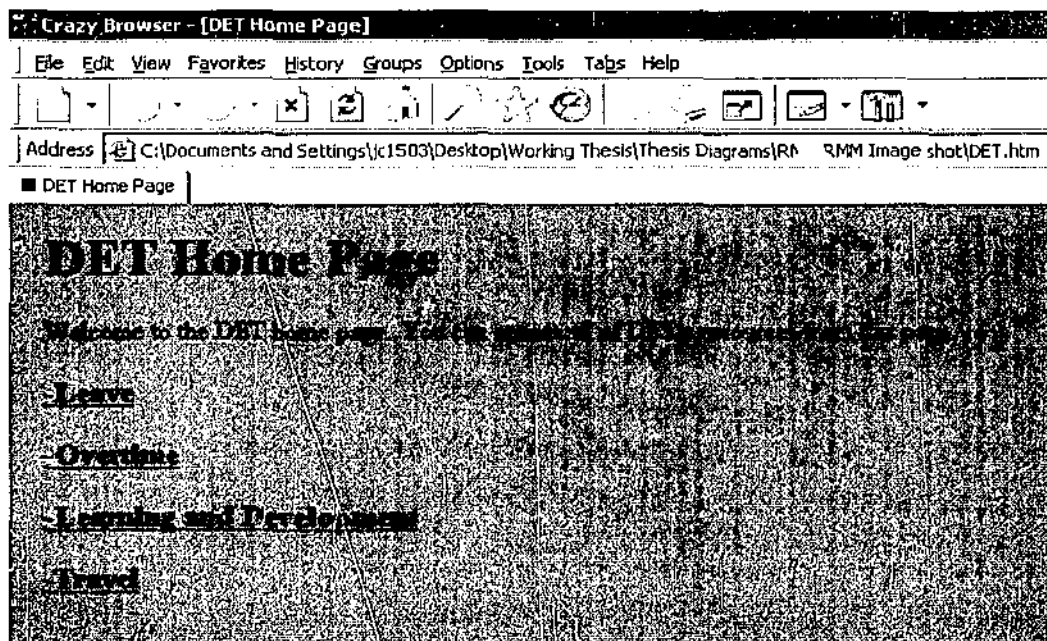
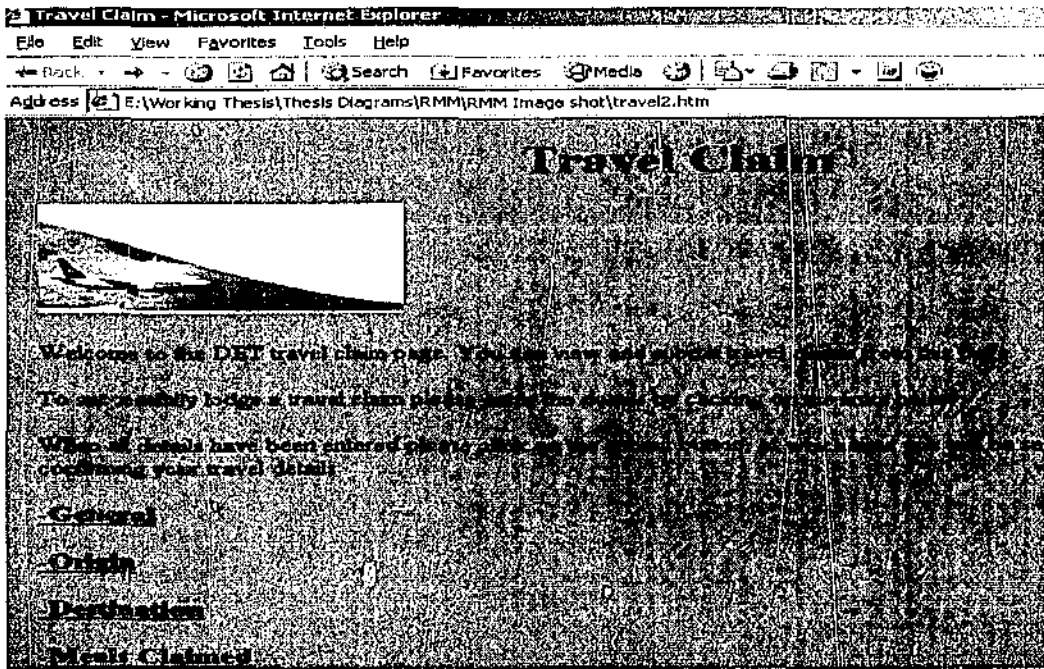


Figure 9.4 – Screenshot of DET Home Page



**Figure 9.5 – Screenshot of Travel page**

On the travel page in figure 9.5, it was decided to use hyperlinks to navigate from the main page to each of the slice subsets i.e. General, Origin, Destination and Meals Claimed. For example, once a user navigates to the origin slice subset, they would enter the details required i.e. travelno and origin and then navigate back to the general slice and click on the submit button to complete the travel claim.

## **10. Analysis of Main Results**

### **10.1 Main Concepts of RMM**

The main concepts of RMM are based on a combination of entity-relationship diagrams and various concepts adapted from Hypermedia Design Methodology (HDM) (Isakowitz et al., 1995). At a first glance, the steps outlined in RMM appear to cover the design phase through to the construction and testing. A deeper investigation reveals that RMM is predominantly focussed on the design phase, particularly the navigation design and although the last four steps are concerned with the physical implementation of the design, very little detail is provided as to how to carry out these steps.

In step 1, the E-R design has the benefit of being a well-documented, widely used approach by systems analysts (Isakowitz et al., 1995). This made the concept of entities and their corresponding relationships easy to model. For certain applications, particularly database driven systems, the E-R diagram may already be developed and can be re-used, making the RMM a desirable option.

In step 2, the Slice Design was a little more difficult to translate as the guidelines provided by Isakowitz et al. (1995) are at best vague in terms of how to slice an entity and which attributes should be kept within a slice and which should be dropped. The main aim of the slice is to eliminate scrolling within a window, but given the increase in the size of the screen real estate over the last couple of years, the designer will need to know the approximate size of the windows to determine how much information can be displayed prior to making slice design decisions.

The concept of the minimal slice may also confuse designers as the attributes assigned to the minimal slice resemble that of a primary key / foreign key relationship found in database designs yet function differently. The relationships between slices of the same entity and external entities were easier to understand and implement, as what was being modelled was the traversal paths between the slices and external entities,



which could be either uni or bi-directional.

The third step, the Navigation Design proved to be intuitive and while there was an initial period of reviewing the RMDM access structures to develop a level of understanding, the translation of the E-R diagram to the E-R+ diagram was relatively easy. Having said this, if there is little understanding of the user requirements and in particular the way in which the users will access the components, the E-R+ diagram could end up being completely wrong. In fact Isakowitz et al. (1995) state that the navigation design is dependant on the requirements analysis and given this is one area that is missing in RMM, there could be a mismatch between user expectation and what is produced in the navigation design, resulting in a re-work of this step.

As previously stated, the last four steps of RMM are only briefly described by Isakowitz et al. (1995). This presented a problem, as there were many ways of interpreting these last four steps. The conversion protocol design as described by Isakowitz et al. (1995) involves producing a set of rules that can be implemented as a program or instructions that developers can follow. This enables the conversion of components of the RMD and E-R diagram into physical objects in the Hypermedia application. While the concept is easy to understand there is no real guidance as to how to undertake this step.

The fifth step, the User Interface Design is concerned with graphically presenting how each object in the RMDM will appear. Again while the concept is easy to understand the lack of procedure leaves the developer not knowing if a particular approach adequately addresses this step. For example, in the DET case, the researcher constructed some basic HTML pages and used a Web browser to assess the layout of the objects. While this may be suitable for small, simple systems, for a complex system how does a developer know if they are using the right font or if the positioning of the objects follows best practice? Should the design be drawn manually on paper by hand or should the user interface be coded? Should the page include video or images? These questions suggest that there may be some problems in carrying out this step.

The last two steps involve designing the run-time behaviour and the conducting the

construction and testing. Both these steps are typical of most system development life cycles, but RMM does not describe the “how to” for these last two steps. Even though the runtime behaviour design and the construction and testing were beyond the scope of this study, it would appear that they occur in parallel, which was not made clear by Isakowitz et al. (1995). Common sense would suggest that to observe the runtime behaviour, the system would need to be in a prototype stage and the behaviour tested in a live situation. Depending on the results the runtime behaviour would be refined until the results required were produced, however this was not made clear nor described in any detail.

## **10.2 Main Concepts of the extended BRD Method**

In the first step of the extended BRD method, the identification of the candidate business rules was made easier as the basic user requirements for SAPI were known beforehand. While the concept of the business rule was easy to understand, the categorisation of the rules into policy, processing or implementation rules confused the issue. Determining which states were externally verifiable was subjective, which caused the researcher to incorrectly categorise a small number of processing rules as policy rules. Implementation rules were not described in great detail and were not the focus of this study so the lack of description did not impact greatly on the study. As more business rules were analysed the category they belonged to became easier to identify and each step in the extended BRD lends itself to adding and refining business rules.

The concept of candidate events and signals was fairly easy to translate. The distinction between an event and a signal was clearly described and the identification of each type, whether it be an event, trigger or message was intuitive. The technique of identifying verbs to identify the events worked well and there was a clear understanding that anything external to the system was a message or trigger. To determine the difference between a message and trigger the researcher simply analysed the events to determine what might trigger the event and what type of message might be sent to the user as a result of the event.

The third step, the identification of candidate objects is not a new concept and the

objects of relevance were easy to identify using Johnstone's (2004) technique of identifying the nouns to identify the objects. Once the objects were identified the next step of identifying the OLHs was simply a case of looking at the possible states an object might occupy and mapping these states. The concept of state transition diagrams is also not new and as a result the OLHs were modelled fairly quickly.

The construction of the UBRDs was initially difficult until a use case diagram was used to model the business rules. It was recognised that by following the business logic of an object moving from one state to another and adding events, conditions and signals that occur during the state transition, the development of the UBRD occurred naturally. The notational constructs used in the UBRDs were easy to understand and implement. The only design construct that was not familiar was the Harel blob, however understanding its use was not difficult.

The BRDs were similar to the UBRDs and apart from numbering the graphic constructs the translation was intuitive. One interesting development was that two of the BRDs that were developed did not conform to the Meta rule described by Johnstone (2004) of state-event-state and had to be refined. Having said this all of the BRDs that were developed were refined using an iterative process to ensure the business rules were being represented accurately. This was then translated into a tabular representation using the ESTs by simply replicating information contained in the BRD.

The construction of the PDDs required additional time to become familiar with the design notations, but the process was made easier as the notations were similar to those used in the BRD. The concept of separating out the processing components from the navigation components was also intuitive and once completed the ESTs were refined to reflect the logic of the PDDs.

The NDDs were developed using the BRDs and the use case diagrams as a reference. Although different design notations are used in the NDDs, the mapping of the navigation paths between nodes was fairly easy and the concept was well understood. The researcher came up with two designs and both proved to be viable solutions, which suggest that a range of solutions is possible and depends on the viewpoint of

the developer.

### **10.3 SDLC Phases Covered by the RMM and the extended BRD Method**

The case study results clearly indicate that the RMM focuses on the design phase of the development lifecycle and only partly covers the implementation phase in the conversion protocol design. There is no requirements analysis and what is interesting to note is that even though steps four to seven are included as part of the method, there is no detailed description of these steps and as such it is questionable if RMM does indeed cover any of the phases in the development lifecycle associated with these steps.

The extended BRD method covers the requirements capture, analysis, conceptual, processing and navigational design phases of the development lifecycle. Johnstone (2004) states that the eBRD method partially covers the implementation phase due to the possible use of embedded XML links in the navigation table, however this was not used in this case study. The testing phase which would occur in parallel with the implementation phase is also not covered, however the extended BRD method does not claim to cover these phases of the SDLC.

### **10.4 Modelling Technique**

The RMM uses the E-R modelling technique, which is well understood and supported by systems developers. The concept of entities and relationships relates well to the Hypermedia domain and the researcher found it easy to design the E-R diagram. The use of slices was a difficult concept to implement, particularly with respect to the construction of slices and where a slice should be used in place of a single entity. The RMDM access structures were also a little difficult to understand and implement but in the end followed a modelling technique not that dissimilar to that used in the E-R diagram, with the relationships between entities replaced by uni or bi-directional links.

In contrast, the extended BRD method is a state-based model, which uses notations

similar to flowcharts (Johnstone, 2004). The use of a flowchart style representation was easy to understand and implement. There was an initial period where familiarisation of the symbols was required, but after this point it was a fairly intuitive process.

The model that was developed using the RMM method was quite different from the model developed using the eBRD method. This can possibly be attributed to the different orientation focus of each method. The RMM focuses on data orientation and the way in which that data is presented depends on its ranking according to its importance within the system. This was particularly evident during the slice design phase. This often led to a de-normalising of the normalised data model to improve navigational efficiency. One drawback to this approach is increased redundancy in the final model.

This focus on data orientation led to a model that was rich in detail with respect to data but lacked sufficient detail for navigation between nodes that did not represent entity-relationship components e.g. privacy statement or feedback page. It also lacked any detail with respect to how the data represented in the model was processed. A possible solution would be to expand the E-R design step by including a separate design that represents the non-entity components which could then be melded in a separate step to more accurately represent the system. The lack of processing detail could also be addressed by refining the navigation design step and follow the approach used in the eBRD by separating out the navigation components and the processing components. This would involve actually identifying the processes and developing a process diagram but the end result would be a more complete representation of the system being constructed.

The eBRD method has a process orientation and therefore the model that was produced represented the processes that occurred within the SAPI system rather than the data that would be manipulated. The separation of the processing elements from the navigational elements and the inclusion of a network orientation in the design phase produced a model that differed in perspective from the RMM model, and one could say, with respect to the navigation design was an enhanced version of the RMM model.

What was interesting was the inclusion of several node documents in the NDD that performed the same function i.e. the confirm details node. This did not translate well to the navigation matrix and warrants further investigation to determine a better way of representing these generic nodes. What is missing from the eBRD method is detail about the data elements which could potentially cause problems with systems that are predominantly data focussed. A solution to this problem may be to incorporate elements of the RMM or another data oriented design method into the eBRD method that handles the data perspective.

### **10.5 Notation and Graphical Representation**

The RMM uses the E-R diagram notations, which are widely used. The researcher found it easy to understand and use the E-R symbols as they have been used in previous projects and are well understood. The RMDM access structures were simple and easy to implement and construction of the diagrams was also easy. The symbols chosen to represent the RMDM access structures tended to reflect the function of the structures quite well. The links between the RMDM access structures were kept simple and were either uni or bi-directional.

The notational symbols that have been chosen for the extended BRD are simple and are not technology specific. Some of the symbols used in the eBRD resemble the symbols used in the E-R diagrams; however care should be taken not to confuse their meaning as they are quite different from those in the E-R diagrams. The symbols used in the PDDs mirrored the symbols used in the BRDs. This made the translation from the BRDs to the PDDs easier as the researcher was not required to become familiar with a new set of symbols.

The notations used in the event specification tables replicated those used in the BRDs and were easy to construct. The NDD notations were also easy to understand and although different from the BRD and PDD notations, were easy to implement. The links between objects was also kept simple and like the links in the RMM, they were either uni- or bi-directional. The use of groups as a way of placing similar nodes together is logical and in a way served the same function as the conditional indexes

used in the RMDM diagram, in that it provided access to all the nodes in that group directly. The Harel blob was used to group together universal links, but was not described as being a construct that could be used in the NDD and as such may warrant further investigation. The navigation table was simply a tabular representation of the NDD and therefore was not difficult to construct, however Johnstone (2004) introduces the concept of embedding XML links in the navigation table which would increase the level of difficulty during the construction of the navigation table.

## 11. Conclusions

Table 11.1 summarises the process, modelling technique, graphical representation and notation used for each of the methods studied closely follows Koch's (1999) design.

**Table 11.1 – Summary of process, modelling technique, graphical representation and notation of RMM and the extended BRD**

	Process	Modelling technique	Graphical representation	Notation
<b>RMM</b>	<ol style="list-style-type: none"> <li>1. E-R design</li> <li>2. Slice design</li> <li>3. Navigational design</li> <li>4. Conversion protocol design</li> <li>5. UI design</li> <li>6. Run-time behaviour design</li> <li>7. Construction and testing</li> </ol>	E-R	<ol style="list-style-type: none"> <li>1. E-R design</li> <li>2. Slice design</li> <li>3. RMDM diagram</li> </ol>	<ol style="list-style-type: none"> <li>1. E-R</li> <li>2. Own</li> <li>3. Own</li> </ol>
<b>eBRD</b>	<ol style="list-style-type: none"> <li>1. Identify candidate business rules</li> <li>2. Identify candidate events and signals</li> <li>3. Identify candidate objects</li> <li>4. Construct OLHs</li> <li>5. Construct UBRDs</li> <li>6. Construct BRDs</li> <li>7. Construct ESTs</li> <li>8. Construct PDDs</li> <li>9. Refine ESTs</li> <li>10. Construct NDDs</li> <li>11. Construct NTs</li> </ol>	<p>State based for the analysis diagrams and the PDD.</p> <p>Network-oriented for the NDD.</p>	<ol style="list-style-type: none"> <li>4. Object states</li> <li>5. Flow chart</li> <li>6. Flow chart</li> <li>7. Table</li> <li>8. Flow chart</li> <li>9. Table</li> <li>10. Navigation diagram</li> <li>11. Matrix</li> </ol>	<ol style="list-style-type: none"> <li>4. Own</li> <li>5. Own</li> <li>6. Own</li> <li>7. Own</li> <li>8. Own</li> <li>9. Own</li> <li>10. Own</li> <li>11. Own</li> </ol>

As can be seen from the summary in table 11.1, both methods propose quite different steps. The RMM focuses almost exclusively on the design phase of the development lifecycle providing quite detailed navigational structures that allow effective navigation and browsing. The RMDM access structures are generic in nature which allows the content to be updated easily. However as Kyaw & Boldyreff (1998) state *"the maintenance for the structure of the application is likely to require that the designers redesign most or part of the existing design"*.



The RMM does not address requirements capture or analysis and does not sufficiently detail steps four to seven, which are concerned with the physical implementation of the system being designed. The E-R modelling technique is widely used and well understood and the graphical representations and the notations used are easy to understand. There is however a lack of detail with respect to processing of the data and non entity components are not given any consideration in the final system.

The extended BRD method is quite intuitive and the steps follow each other logically and allow for continuous refinement during each phase. There is a detailed requirements capture and analysis step that allows developers to accurately map the business requirements of a system. The navigation design sufficiently captures the paths a user may use to navigate within the system to be designed, but may not be as detailed as the RMDM access structures used in the RMM with respect to data detail.

Table 11.2 lists the strengths and weaknesses identified for both methods used in the DET case.

**Table 11.2 – Strengths and Weaknesses of the RMM and eBRD Method**

<b>Method</b>	<b>Strengths</b>	<b>Weaknesses</b>
<b>RMM</b>	E-R modelling technique is well understood.	Restricted to specific types of applications.
	Well suited to database oriented systems.	Focuses only on the design phase in the development lifecycle with no requirements capture or analysis.
	Well suited to systems with well defined data structures and information with a high level of volatility.	Lack of process perspective.
<b>eBRD</b>	Modelling technique is simple.	Lack of data perspective.
	Effective user requirements capture.	Limited field testing so far (two studies conducted by McDermid and four studies conducted by



## **12. Further Research Work**

While the single case design has served its purpose in this research, an avenue for future work is the application of both methods using a multiple case design. Yin (2003, p46) states that there are distinct advantages using a multiple case design. Using multiple cases allows for more evidence collection and therefore a more robust result. Application of both the RMM and the extended BRD method to the development of different systems with different user requirements would enable a more detailed comparison. The replication of the results achieved in this study across a number of cases would also provide more substantial support for the initial proposition.

Another area for future research is investigation of the use of the Harel blob to represent global nodes in the NDD. This happened quite by accident and was used as a means of reducing the number of links that required to be represented with respect to the global nodes in the system. The use of generic nodes that perform the same function for certain parts of the system is also of interest and the way in which these nodes are represented diagrammatically in the NDD should be explored.

### **13. Concluding Remarks**

In closing, this thesis analysed two methods that could be applied to the development of a We-based hypermedia system. It specifically looked at the phases covered by both methods in the development life-cycle in an effort to ascertain how effective they were at translating a real world problem. In conducting this work it has become apparent that additional studies are required to further validate the results of this research and build upon the existing literature. Having said this, the research conducted in this thesis has satisfied its original aims and posed questions that require further investigation.

## 14. References

- Alavi, M. and Carlson, P. (1992) A Review of MIS Research and Disciplinary Development, *Journal of Management IS*, (8:4), pp. 45-62.
- Balasubramanian, V, Bieber, M, Isakowitz, I (1996) Systematic Hypermedia Design, *IS*, 26(4), pp. 295-320.
- Barna, P., Frasincar, F., Houben, G., Vdovjak, R. (n.d) *Methodologies for Web Information System Design* [on-line]. Available WWW: [www.wis.win.tue.nl/~hera/papers/ITCC2003b/itcc2003b.pdf](http://www.wis.win.tue.nl/~hera/papers/ITCC2003b/itcc2003b.pdf) accessed 17/05/2004 at 10:11am
- Barry, C. & Lang, M. (2001) A survey of multimedia and Web development techniques and methodology usage, *IEEE Multimedia*, 8(2), pp. 52-60
- Benbasat, I., Goldstein, D. and Mead, M. (1987) The Case Research Strategy in Studies of IS, *MIS Quarterly* (11:3) pp. 369-385.
- Berk, E. and Devlin, J. (1991). *Hypertext/Hypermedia Handbook*. New York: McGraw-Hill Publishing Company, Inc.
- Bichler, M. and Nusser, S (n.d.) *Modular Design of Complex Web-Applications with SHDT* [on-line]. Available WWW: <http://www.wicky.wu-wein.ac.at/w3dt/wetice/wetice.html> accessed 02/09/2003 at 10:30am
- Bieber, M. and Vitali, F. (1997). Toward Support for Hypermedia on the World Wide Web. *IEEE Computer* 30(1): 62-70.
- Brusilovsky, P. (1998) *Methods and techniques of adaptive hypermedia* Netherlands: Dordrecht: Kluwer Academic Publishers
- Buckingham Shum, S. & McKnight, C. (1997) World Wide Web usability: introduction to this special issue, *International Journal of Human-Computer Studies*, 47(1), 1-4.
- Bush, V. (1945). As We May Think. *The Atlantic Monthly* 176.1 (July) p. 101-103.
- Campbell, D and Stanley, J. (1966) *Experimental and Quasi-Experimental Designs for Research*. Chicago: Rand McNally.
- Campbell, D. (1975) Degrees of freedom and the case study, *Comparative Political Studies*, 8, 178-193.
- Chen, P.P. (1976). The entity-relationship model: toward a unified view of data, *ACM Transactions on Database Systems*, 1:1 pp 9-36.

- Costagliola, G., Ferrucci, F., Francese, R. (n.d) *Web Engineering: Models and Methodologies for the Design of Hypermedia Applications* [on-line]. Available WWW: [www ldc.usb.ve/~abianc/electivas/WebEngineering.pdf](http://www ldc.usb.ve/~abianc/electivas/WebEngineering.pdf) accessed 17/05/2004 at 10:05am
- Darke, P and Shanks, G (2000). Case study research, In Williamson, Kirsty (2000). *Research methods for students and professionals*. Wagga Wagga, N. S. W.: Centre for Information Studies, pp.93-105.
- Diamond, J. (1996) The roots of radicalism. *The New York Review of Books*, 14 November, pp. 4–6.
- Dogan, M and Pelassy, D (1990) *How To Compare Nations: Strategies in Comparative Politics (2nd ed.)*. London: Chatham House.
- Dolog, P., Gavrioloaie, R., NejdI, W., Brase, J. (2003) Integrating Adaptive Hypermedia Techniques and Open RDF-Based Environments, *Proceedings of the Twelfth International World Wide Web Conference*, Budapest, Hungary
- Eklund, J. and Lowe, D. (2002) *Using Partial Designs to Elicit Requirements in Web Development – A Survey of Commercial Practice* [on-line]. Available WWW: <http://ausweb.scu.edu.au/aw02/papers/edited/eklund/paper.html> accessed 13/05/2004 at 12:25pm
- Feizabadi, S. (1998) *History of the World Wide Web* [on-line]. Available WWW: <http://ei.cs.vt.edu/~wwwbtb/book/chap1/index.html> accessed 06/04/2004 at 9:30pm
- Fitzgerald, B. (1997) The Use of Systems Development Methodologies in Practice: A Field Study. *IS Journal*. 7(3), 201-212.
- Flyvbjerg, B. (2003) *Five Misunderstandings About Case Study Research* [on-line]. Available WWW: [www.plan.aau.dk/~flyvbjerg/MSFiveMis9.0SageASPUBL.pdf](http://www.plan.aau.dk/~flyvbjerg/MSFiveMis9.0SageASPUBL.pdf) accessed at 1:42 pm on 09/06/2004
- Gallagher, S. & Webb, B. (1997) Competing Paradigms in Multimedia Systems Development: Who Shall Be The Aristocracy? In *Proceedings of Fifth European Conference on IS*, Cork, Ireland, June 19-21 1997.
- Gross, N., et al. (1971). *Implementing organisational innovations*. New York: Basic Books.
- Garzotto, F., Paolini, P. and Schwabe, D. (1993) HDM – A Model-based Approach to Hypertext Application Design, *ACM Transactions on IS*, 11(1), 1-26.
- Gromov, G (2003) *History of Internet and WWW: View from Internet Valley* [on-line]. Available WWW: <http://www.netvalley.com/intval1.html> accessed 12/05/2004 at 11:15am
- Harel, D. (1988) On Visual Formalisms, *Communications of the ACM*, 31(5), 514-30.

- Henderson-Sellers B. (1995). Who needs an object-oriented methodology anyway? *Journal of Object Oriented Programming*, Vol 8(6), 6-8.
- Interactivity in Motion (2002) *Aspen Movie Map* [on-line]. Available WWW: [http://seminare.design.fh-aachen.de/inmotion/stories/storyReader\\$17](http://seminare.design.fh-aachen.de/inmotion/stories/storyReader$17) accessed 10/04/2004 at 10:37am
- Isakowitz, T (1997) *What is RMM* [on-line]. Available WWW: [http://jmis.bentley.edu/rmm/What\\_is\\_RMM.html](http://jmis.bentley.edu/rmm/What_is_RMM.html) accessed 9/05/2004 at 4:18pm
- Isakowitz, T., Stohr, E. A., and Balasubramanian, P. (1995) RMM: A Methodology for Structured Hypermedia Design, *Communications of the ACM*, 38(8), 34-44.
- Johnson, P. & Nemetz, F. (1998) Towards principles for the design and evaluation of multimedia systems. *Human-Computer Interaction (HCI'98)*, Sheffield, England, Springer-Verlag
- Johnstone, M.N., and McDermid, D.C. (1999). 'Extending and Validating the Business Rules Diagram Method'. *Pages 449-461 of: Hope, B. and Yoong, P. (eds), Proceedings of the 10th Australasian Conference on IS*, Wellington, New Zealand
- Johnstone M.N., McDermid D.C. and Venable J.R., (2002) "Modelling E-business Security Requirements: Developer and Client Expectations", In: Cybulski, J.L., Nguyen, L., Lamp, J., and Smith, R. (eds), *Proceedings of the Seventh Australian Workshop on Requirements Engineering*, CDROM, pp111-122
- Johnstone M.N., McDermid D.C. and Venable J.R. (2003). "Extending the Business Rules Diagram Method to Web Based System Design" In: Ang, J. and Knight, S-A. (eds), *Proceedings of the 14th Australasian Conference on IS*, CD-ROM.
- Johnstone, M. N. (2004) Validating and Extending the Business Rules Diagram Method *PhD thesis to be submitted Curtin University of Technology, Perth, Western Australia*
- Keen, P. G. W. (1991) Relevance and Rigor in IS Research: Improving Quality, Confidence, Cohesion and Impact. In: *IS Research: Contemporary Approaches and Emergent Traditions*, pp. 27-49. Elsevier Science Publishers (North-Holland).
- Klein, H. and Myers, M. (1999) A Set of Principles for Conducting and Evaluating Interpretive Field Studies in IS, *MIS Quarterly*, (23:1), pp. 67-94.
- Koch, N (1999) A Comparative Study of Methods for Hypermedia Development. *Technical report*, LMU 9905, Ludwig-Maximilians-Universität München Institute of Computer Science
- Kyaw, P & Boldyreff, C (1998) *A survey of Hypermedia design methods in the context of World Wide Web design* [on-line]. Available WWW: <http://citeseer.nj.nec.com/kyaw98survey.html> accessed 05/10/2003 at 8pm

- Lang, M (2001) *A Study of Practice in Hypermedia Systems Design* [on-line]. Available WWW: [ecis2001.fov.uni-mb.si/doctoral/ Students/ECIS-DC\\_Lang.pdf](http://ecis2001.fov.uni-mb.si/doctoral/Students/ECIS-DC_Lang.pdf) accessed 17/05/2004 at 8:30am
- Lang, M (2002) *Hypermedia Systems Development-Do We Really Need New Methods?* [on-line]. Available WWW: <http://ecommerce.lebow.drexel.edu/eli/2002Proceedings/papers/Lang148Hyper.pdf>
- Lee, A. S. (1989) A Scientific Methodology for MIS Case Studies, *MIS Quarterly*, (13:1), 32, pp. 33-52
- Lee, H., Lee, C., and Yoo, C. (1999) A Scenario-based Object-oriented Hypermedia Design Methodology, *Information & Management*, 36, 121-38.
- Linden, T and Cybulski, J (2003) *Solving Common Problems in Multimedia Design* [on-line]. Available WWW: [http://www.deakin.edu.au/infosys/docs/workingpapers/papers/2003\\_20\\_Linden.pdf](http://www.deakin.edu.au/infosys/docs/workingpapers/papers/2003_20_Linden.pdf)
- Lowe, D. (1999). *Engineering the Web* [on-line]. Available WWW:[http://www.aace.org/pubs/webnet/v1no3/lowe1\\_3.pdf](http://www.aace.org/pubs/webnet/v1no3/lowe1_3.pdf) accessed 07/09/2003 at 3:40pm
- Lowe, D & Henderson-Sellers, B. (n.d.) *Characteristics of Web Development Processes* [on-line]. Available WWW: [http:// www.ssgrr.it/en/ssgrr2001/papers/David%20Lowe.pdf](http://www.ssgrr.it/en/ssgrr2001/papers/David%20Lowe.pdf) accessed 07/09/2003 at 3:30pm
- Lowe, D., Bucknell, A. & Webby, R (1999) *Improving Hypermedia Development: A Reference Model-Based Process Assessment Method* [on-line]. Available WWW: [http://www.google.com.au/search?q=cache:V9\\_RmCYVoVMJ:www.eng.uts.edu.au/~dbl/papers/99ht01.pdf](http://www.google.com.au/search?q=cache:V9_RmCYVoVMJ:www.eng.uts.edu.au/~dbl/papers/99ht01.pdf) accessed 15/05/2004 at 10:15am
- Markus, M. (1997) *The Qualitative Difference in IS Research and Practice*. London: Chapman and Hall.
- McDermid, D. C. (1998) The Development of the Business Rules Diagram, *PhD thesis* Curtin University of Technology, Perth, Western Australia.
- Naimark, M (1997) A 3D Moviemap and a 3D Panorama. *SPIE Electronic Imaging Proceedings* Vol. 3012, pp 169-179
- Nanard J. and Nanard M. (1995). Hypertext design environments and the hypertext design process. *Communication of the ACM*, August 1995, Vol 38 (8), 49-56.
- Nelson, T. (1990) *Literary Machines*. The Distributors
- Nielsen, J. (1988) Hypertext '87 Trip Report. *ACM SIGCHI Bulletin* 19, 4, pp.27-35
- Nielsen, J. (1990) *Hypertext and Hypermedia*. Academic Press, Boston



Nykanen, O. (n.d) *Digital Hypermedia Technologies* [on-line]. Available WWW: [http://matwww.ee.tut.fi/~onykane/papers/digital\\_hmt/dht.html](http://matwww.ee.tut.fi/~onykane/papers/digital_hmt/dht.html) accessed 12/05/2004 at 3:10 pm

Orlikowski, W. J. and Baroudi, J. J. (1991) Studying IT in Organizations: Research Approaches and Assumptions, *IS Research*, (2:1), pp. 1-28.

Paré, G. and Elam, J. J. (1997) Using Case Study Research to Build Theories of IT Implementation, in *IS and Qualitative Research*, A. S. Lee, J. Liebenau, and J. I. DeGross (eds.), Chapman & Hall, pp. 542-568.

Paynter, J & Pearson M (1998). *A Case Study of the Web Based IS Development* [on-line]. Available WWW: [http://www.cecil.edu/reports/www\\_Case\\_Multimedia\\_98.pdf](http://www.cecil.edu/reports/www_Case_Multimedia_98.pdf) accessed 03/10/2003 at 2:15pm

Relihan, L., Cahill, T. & Hinchey, M (1994) Untangling The World Wide Web, *Proceedings ACM SIGDOC'94*, Banff, October 1994, pp 17-24

Robey, D. and Newman, M. (1996) Sequential Patterns in IS Development: An Application of a Social Process Model, *ACM Transactions on IS*, (14:1), pp. 30-63.

Rumbaugh J. (1995). What is a method? *Journal of Object Oriented Programming*, Vol 8(6), 10-16.

Russell, P. (2000) Infrastructure - Make or Break your E-Business. In *TOOLS-Pacific 2000: Technology of Object-Oriented Languages and Systems*, Sydney, Australia

Sarker, S. and Lee, A. S. (2000) Using A Case Study To Test The Role of Three Key Social Enablers in ERP Implementation, *Proceedings of the International Conference on IS*, Brisbane, Australia.

Schramm, W. (1971) Notes on case studies of instructional media projects. *Working paper for the Academy for Educational Development*, Washington, DC.

Schwabe, D. and Rossi, G. (1995) The Object-Oriented Hypermedia Design Model, *Communications of the ACM*, 38(8), 45-46.

Schwabe D., Rossi G. and Barbosa S. (1996). Systematic hypermedia design with OOHDM. In *Proceedings of the ACM International Conference on Hypertext, Hypertext '96*.

Takahashi, K. and Liang, E. (1997) Analysis and design of Web-based IS. *Computer Networks and ISDN Systems* v.29 n.8-13, p.1167-1180.

Tellis, W (1997) *Results of a Case Study on Information Technology at a University. The Qualitative Report*. [on-line serial], 3(4). Available WWW: <http://www.nova.edu/ssss/QR/QR3-3/tellis3.html> accessed 01/09/2004 at 9:53 am

Thomas, D. (1998) Web Time Software Development. *Software Development Magazine*. October, 78-80.

Webopedia (2003) [on-line]. Available WWW: <http://www.pcwebopedia.com/> accessed 12/05/2004 at 10:30am

W3C (n.d) *Ted Nelson and Xanadu* [on-line]. Available WWW: <http://www.w3.org/Xanadu.html> accessed 12/05/2004 at 10:00am

W3C (2000) *A Little History of the World Wide Web* [on-line]. Available WWW: <http://www.w3.org/History.html> accessed 12/05/2004 at 10:15am

W3C (2003) *Some early ideas for HTML* [on-line]. Available WWW: <http://www.w3.org/MarkUp/historical> accessed 08/10/2003 at 11:25am

Wynekoop J. L and Russo N.L. (1993) Systems Development Methodologies; unanswered questions and the research-practice gap. *14th International Conference on IS*. Orlando Florida, 1993.

Wynekoop J. L and Russo N.L. (1997) Studying System development methodologies: an examination of research methods. *IS Journal*. 7, pp. 47-65.

Yin, R. (1993) *Applications of Case Study Research* Beverly Hills: Sage Publications, Inc

Yin, R. (2003) *Case Study Research -- Design and Methods*, California: Sage Publications, Inc.