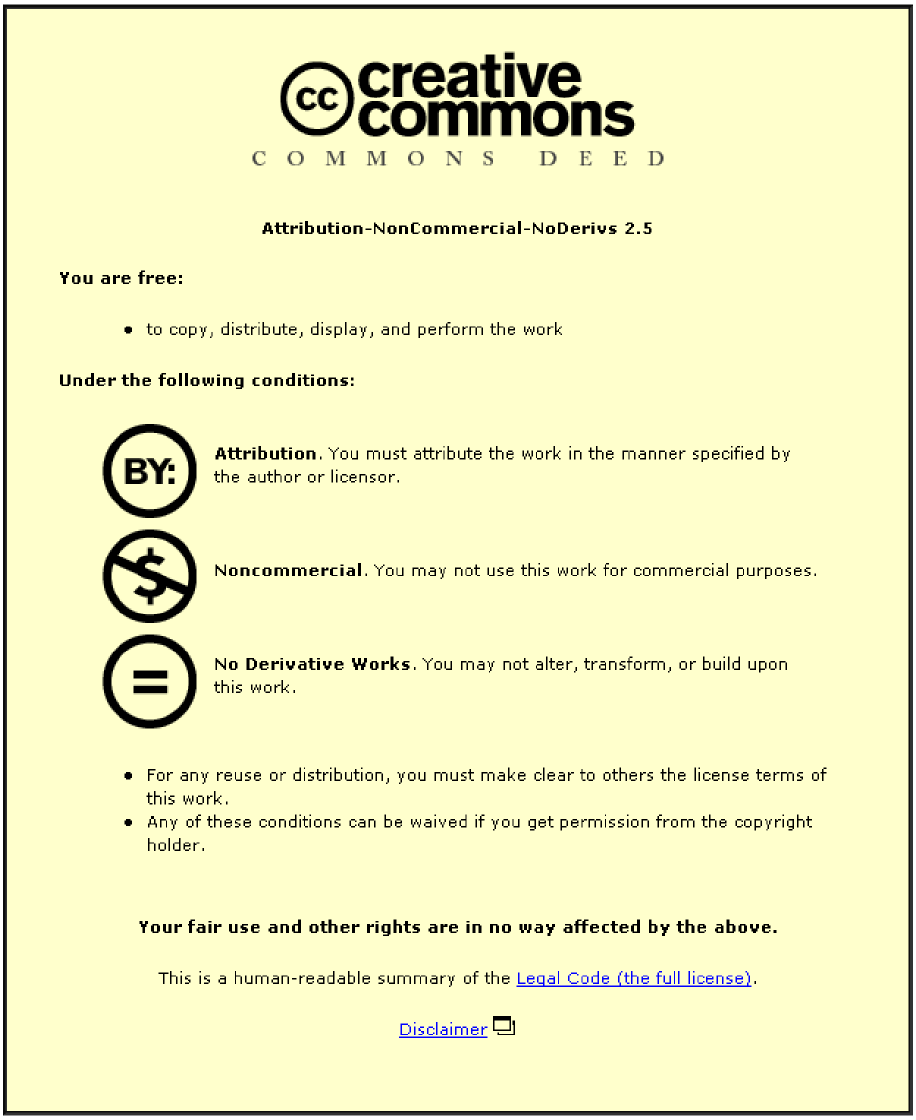


HYBRID SEMANTIC-DOCUMENT MODELS

By

Darren Clowes

A Doctoral thesis submitted in partial fulfilment of the requirements for the award of the degree Doctor of Engineering (EngD), at Loughborough University

December 2013

BAE Systems
Military Air Solutions (MAS)
Warton Aerodrome
Preston
Lancashire, PR4 1AX

Centre for Innovative and Collaborative Engineering
Department of Civil & Building Engineering
Loughborough University
Loughborough
Leicestershire, LE11 3TU

[This page intentionally left blank.]

ABSTRACT

This thesis presents the concept of hybrid semantic-document models to aid information management when using standards for complex technical domains such as military data communication. These standards are traditionally text based documents for human interpretation, but prose sections can often be ambiguous and can lead to discrepancies and subsequent implementation problems. Many organisations produce semantic representations of the material to ensure common understanding and to exploit computer aided development. In developing these semantic representations, no relationship is maintained to the original prose. Maintaining relationships between the original prose and the semantic model has key benefits, including assessing conformance at a semantic level, and enabling original content authors to explicitly define their intentions, thus reducing ambiguity and facilitating computer aided functionality.

Through the use of a case study method based on the military standard MIL-STD-6016C, a framework of relationships is proposed. These relationships can integrate with common document modelling techniques and provide the necessary functionality to allow semantic content to be mapped into document views. These relationships are then generalised for applicability to a wider context. Additionally, this framework is coupled with a templating approach which, for repeating sections, can improve consistency and further enhance quality. A reflective approach to model driven web rendering is presented and evaluated. This reflective approach uses self-inspection at runtime to read directly from the model, thus eliminating the need for any generative processes which result in data duplication across source used for different purpose.

KEY WORDS

Meta-Models, Document Models, Document Engineering, Standards, Conceptual Models, Usability

ACKNOWLEDGEMENTS

I would like to thank Prof. Ray Dawson for making me aware and giving me the opportunity of undertaking this Engineering Doctorate project. I would like to thank him and Dr. Steve Probets for their support, supervision and guidance of the highest level.

Dr. Julian Johnson acted as the industrial supervisor, without who this project would not have been possible. His direction, knowledge and wisdom were of great benefit. Dr. Chris Holmes of BAE Systems developed a key aspect for this system in the form of the semantic structure models. His domain and modelling knowledge and experience was most valuable and his assistance most appreciated. I would also like to thank Dr. Ben Gory and Dr. John Rowlands of BAE Systems who during the course of the project acted as Industrial Coordinators. Their oversight, support and insightful wisdom were much appreciated.

Special thanks are due to the Epsilon team at the University of York, especially Dimitris Kovolos, for their help and support in aiding with their suite of modelling languages which were heavily used in the project. I would like to thank all the members of staff at the Systems Engineering Innovation Centre at Loughborough, who created such a great and enjoyable office environment to work in. I wish all of you well in your futures.

And finally, I would like to thank my family for their continued support during this period of study, especially my wife Catherine for standing by me during years of academic study, I love you lots.

PREFACE

This thesis forms part of the submission requirement for the Engineering Doctorate (EngD) research programme administered by the Centre for Innovative Collaborative Engineering (CICE) at Loughborough University. The research was sponsored by the UK defence firm BAE Systems.

The Engineering Doctorate (EngD) is an EPSRC sponsored alternative to a traditional PhD being better suited to industry needs and vocationally oriented. The EngD is focused on industrial innovation and as such an objective is for innovation in the application of knowledge rather than simply a contribution to the body of knowledge. About 75% of the programme time is spent working directly with a company undertaking a commercially relevant research project or portfolio of projects. The programme is four-years and combines PhD-level research projects with masters-level taught courses.

USED ACRONYMS / ABBREVIATIONS

AFAPD	Air Force Application Protocol Development
AJAX	Asynchronous JavaScript and XML
APIS	Actual Platform Implementation Specification
APR	Apache Portable Runtime
ATL	ATLAS Transformation Language
BREAD	Browse, Read, Edit, Add and Delete
CDA	Clinical Document Architecture
CDF	Computable Document Format
CPU	Central Processing Unit
CRUD	Create, Read, Update and Delete
DFI	Data Field Identifier
DITA	Darwin Information Typing Architecture
DLCP	Data Link Change Proposals
DoD	Department of Defense
DOORS	Dynamic Object-Oriented Requirements System
Dstl	Defence Science and Technology Laboratory
DTD	Document Type Definition
DUI	Data Use Identifier
ECL	Epsilon Comparison Language

EGL	Epsilon Generation Language
EMDA	East Midlands Development Agency
EMF	Eclipse Modelling Framework
EngD	Engineering Doctorate
EOL	Epsilon Object Language
EPSRC	Engineering and Physical Sciences Research Council
EPUB	Electronic Publication
ETL	Epsilon Transformation Language
GMT	Generative Modeling Technologies
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IIS	Internet Information Services
JSP	JavaServer Pages
JTIDS	Joint Tactical Information Distribution System
JVM	Java Virtual Machine
M2T	Model To Text
MAS	Military Air Solutions
MDR	MetaData Repository
MDWE	Model Driven Web Engineering
MIDS	Multifunctional information distribution system

MIF	Maker Interchange Format
MIL-STD	Military Standard
MOF	Meta-Object Facility
MS	Microsoft
NATO	North Atlantic Treaty Organization
NDD	National Difference Document
NEC	Network Enabled Capability
NIO	New I/O
NRS	National Requirements Specification
OASIS	Organization for the Advancement of Structured Information Standards
OCL	Object Constraint Language
ODBMS	Object Database Management Systems
OMG	Object Management Group
OODBMS	Object-Oriented Database Management Systems
OO-H	Object-Oriented Hypermedia
OOWS	Object Oriented Web Solutions
OOXML	Office Open XML
ORDBMS	Object Relational Database Management Systems
ORM	Object-Relational Mapping
PDF	Portable Document Format

PHP	PHP: Hypertext Preprocessor
PIDD	Platform Implementation Difference Document
PRDD	Platform Requirements Difference Document
PRS	Platform Requirements Specification
QVT	Query/View/Transformation
RAM	Random-access Memory
RDBMS	Relational Database Management Systems
RIM	Reference Information Model
RTF	Rich Text Format
SDD	Service Difference Document
SEIC	Systems Engineering Innovation Centre
SQL	Structured Query Language
SRS	Service Requirements Document
STANAG	Standardization Agreement
TADIL	Tactical Digital Information Link
TDL	Tactical Data Link
TDMA	Time Division Multiple Access
TEI	Text Encoding Initiative
TES	Technology and Engineering Services
UAV	Unmanned Aerial Vehicle

UML	Unified Modeling Language
URL	Uniform Resource Locator
UWE	UML-based Web Engineering
VMF	Variable Message Format
WebML	Web Modeling Language
WEI	Web Engineering Interoperability
XHTML	Extensible HyperText Markup Language
XMF	Executable Meta-modelling Framework
XMI	XML Metadata Interchange
XML	Extensible Markup Language

GLOSSARY

ATL ATL is a model transformation language and toolkit

DITA DITA is an XML data model for authoring documents

DocBook DocBook is a semantic markup language for technical documentation

Eclipse Eclipse is an open source platform for the development of highly integrated tools and rich client applications.

Ecore Ecore is the meta-model for describing models with the Eclipse Modelling Framework

EMF The EMF project is a modelling framework and code generation facility for building tools and other applications based on a structured data model in the Eclipse platform

Epsilon Epsilon is a component of the Eclipse Modelling GMT project that provides tools and domain-specific languages for Model-Driven Engineering

Mandril Mandril is an integrated suite of software to interpret and analyse message data flows within Tactical Data Links

MDWE MDWE is the philosophy of applying model driven engineering principes to web development

MIDAS MIDAS is an approach for Web development utilising model driven engineering principles

OCL OCL is a declarative language for describing rules that apply to UML models

OO-H OO-H is an approach for Web development utilising model driven engineering principles

QVT	QVT is a set of languages for model transformation
DOORS	DOORS is a requirements management application for optimising requirements communication, collaboration and verification
TEI	TEI is a standard for the representation of texts in digital form
UWE	UWE is an approach for Web development utilising model driven engineering principles
XMap	XMap is a language used by XMF for generative transformations
XMF	XMF is a programming language for language oriented programming and developing domain specific languages

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF PAPERS

The following papers, included in the appendices, have been produced in partial fulfilment of the award requirements of the Engineering Doctorate during the course of the research.

PAPER 1 (SEE APPENDIX A)

Clowes, D., Dawson, R. & Probets, S. (2013). Extending document models to incorporate semantic information for complex standards. *Computer Standards & Interfaces*.36 (1). pp.97-109.

PAPER 2 (SEE APPENDIX B)

Clowes, D., Kolovos,D., Holmes, C., Rose, L., Paige, R., Johnson, J., Dawson, R. & Probets, S. (2010) A reflective approach to model-driven web engineering. In Kühne, T., Selic, B., Gervais, M.P. & Terrier, F., eds, *Modelling Foundations and Applications*, volume 6138 of *Lecture Notes in Computer Science*, Springer Berlin/Heidelberg, pp. 62–73.

PAPER 3 (SEE APPENDIX C)

Clowes, D., Dawson, R., Probets, S., Johnson, J. & Holmes, C. (2008), Pilot studies in using the semantic knowledge of information in large technical documents to aid user navigation, in Harorimana, D. & Watkins,D. eds, *'The 9th European Conference on Knowledge Management: Eckm 2008'*, Academic publishing, pp. 129–138.

PAPER 4 (SEE APPENDIX D)

Clowes, D., Dawson, R., Probets, S. & Johnson, J. (2009), Quality interaction with database driven large documents over HTML. In Dawson, R., Ross, M., & Staples, G., eds, *Software Quality in the 21st Century*, The British Computing Society, pp. 195–206.

PAPER 5 (SEE APPENDIX E)

Clowes, D., Johnson, J., Dawson, R. & Probets, S.. (2008) Quality issues of user interaction in large documents. In Uhomoibhi, J., Dawson, R., Ross, M., and Staples, G., eds, *Quality Issues for Business*, British Computing Society, pp. 37–48.

ADDITIONAL PAPERS (NOT INCLUDED IN THIS THESIS)

Clowes, D., Dawson, R., Probets, S. & Johnson, J., (2010) Adding quality to web development. In Dawson, R., Ross, M. & Staples, G., eds, *Quality Issues for Today*, The British Computing Society, pp. 123–134.

Clowes, D., Dawson, R., Probets, S., Johnson, J., Holmes, C. & Ajit, S. (2009) A multi-layered approach to the development of custom tactical data link applications. In *Proc. of Conference on Systems Engineering Research 2009*.

INDUSTRIAL PRESENTATIONS (NOT INCLUDED IN THIS THESIS)

Invited Speaker: Dassault Systemes UK Product Lifecycle Management Forum 2011. Chelsea, United Kingdom 12th May 2011.

Invited Speaker: IQPC Tactical Data Links 2011: Delivering Real Time Situational Awareness. Prague, Czech Republic 23-24th February 2011.

1 INTRODUCTION

CHAPTER PREFACE

This chapter places the research into context by introducing the environment in which the research was conducted. An overview of the domain, complex engineering standards, is provided, as well as background to the research sponsor BAE Systems and the case study domain Tactical Data Links. The research aims and objectives are identified and a thesis outline provides a précis of the main chapters.

1.1 SUBJECT DOMAIN

Increasingly the design of complex engineered products and systems (herewith referred to as systems) are becoming more reliant on computer-supported models/representations of information, which can be used with computerised checking algorithms to ensure consistency and correctness. Computer-supported models enable engineers to represent information graphically, which can aid understanding; these models also give a more rigorous definition of the systems requirements than that of the more ambiguous (open to interpretation) prose representation of the same information. Many system properties can be modelled, such as behaviour, functionality, verification information and manufacturing instructions.

In addition, some systems have long lifecycles, for example the defence and aerospace industries have systems with a 40+ year lifespan. As such, there is a lot of active legacy information, which has been written without consideration or thought towards being modelled. However, in the complex engineering domain, even legacy documents should represent rigorous elements of a design, but the degree of rigour may have been limited by the available formalisms at the time they were created.

Given that documents within both a new system and a legacy system could contain aspects that can and cannot be modelled, the hypothesis was posed that if these two aspects were combined, they could add additional benefit to users. As such, the proposal of the research topic Hybrid Semantic-Document Models evolved. The aim, for documents with both aspects (formal, semantic or modelled, and informal) was to gain advantages from both the standard document structure and the benefits of a semantic model of the information, resulting in a hybrid semantic-document technique.

1.1.1 COMPLEX ENGINEERING DOMAINS

Why only complex engineering domains? For the purpose of this research a complex engineering domain refers to a domain where the documented materials are large in volume and contain rigorous elements that could be modelled semantically. This differs from non-complex domains which are characterised by one of the following:

- The documentation contains/requires little or no rigour
- The document contains rigour but little benefit would be gained from the overhead of a semantic model
- The volume of information is small, such that its current representation (likely a structured document) is sufficient to have negligible impact on its use by engineers.

In engineering domains, the documentation is normally highly structured. Engineers are required to use, and have a detailed knowledge of these documents to perform their task. In complex engineering domains, these documents normally consist of large sections of prose. It can, therefore, be hard to understand the precise objective of the section. These sections generally contain elements of rigour, which could be used semantically, and more ambiguous prose, which helps the reader’s understanding. These documents are also often only available in the form in which they were written, e.g. a hard copy printout or an electronic PDF or Word format. Some may even be available as a HTML page.

In most cases the document producer has not considered how the end engineer is going to use the document. It is written with the primary focus of acting as an information dump for the author. For small documents (a couple of hundred pages) this static representation may not prove inconvenient. However, imagine the inconvenience of having a document that is several thousand pages in size being used by an engineer who is trying to track the symptoms they are receiving from testing to find an error. The engineer in this case is likely to be jumping around the document, constantly having to revisit the index to find the next appropriate section. Engineers have become accustomed to using such documents. However, many of these documents are highly structured, which enables efficient indexing and searching, but what is often ignored is that, because of the domain, the documents also contain significant elements of rigour which can be used to create a semantic representation of the document as well. For example, a document describing the standard for some communication link is likely to be structured with headings such that the possible messages that can be sent are all listed and discussed. However, in the discussion of the message or at a later point of the document, it may be defined that messages may only be allowed to occur in a fixed order. This order is the semantic knowledge, which could lead to a semantic representation of the elements referenced in the document.

There are many areas of the complex engineering domain that could benefit from the use of the semantic and static representation of information. The types of domains that will benefit include those that have associated standards documents. The defence industry contains many such documents; other industries that have similar documents, which this technique may benefit, include the automotive, aerospace, nuclear and legislative sectors.

1.1.2 WHAT ARE RIGOUR AND SEMANTICS?

In the previous introductory sections, a lot of emphasis is placed on the terms rigour and semantics; what is actually meant by these terms? Both terms are interlinked; rigour is used to describe elements of information that are clearly defined and unambiguous in meaning. Rigorous elements leave no possibility for different interpretations and, as such, they can be modelled to create the semantic information.

Semantic information is the formalised description of the meaning and relationship between elements. This information is often represented in a semantic model. The semantics being fixed due to the rigorous nature can then be considered as objects that can be processed by computerised checking algorithms. An example of semantics is that of a message construct with a communication standard such as that described in the next section. From the standard it could be determined that particular messages must contain certain bits (binary digits), or can only have certain values. This knowledge can be used to create the semantic model that can then be used to check that all messages meet these rigorous standards.

1.2 RESEARCH CONTEXT

1.2.1 THE SPONSOR

This project was funded jointly by EPSRC and BAE Systems through their Military Air Solutions (MAS) business unit. As an Engineering Doctorate project, it was primarily funded through the EPSRC with the industrial sponsor providing the additional funding beyond the EPSRC stipend.

At the time of writing BAE Systems was the UK's largest defence contractor and was a global defence, security and aerospace company with approximately 107,000 employees worldwide. The company had a wide ranging portfolio of products in all sectors of defence (air, land and naval forces), from state of the art aircraft like Typhoon, to the very latest nuclear attack submarine, Astute. It also offered advanced electronics, security, information technology solutions and customer support services. It had customers in over 100 countries with its 2009 sales exceeding £22.4 billion. The company operated in seven home markets - Australia, India, Saudi Arabia, South Africa, Sweden, UK and the US, with major business units based in the UK, US and Australia. The depth, breadth and complexity of engineering capability at BAE Systems was unique, as the company managed some of the most complex, innovative and challenging projects in the world today. It employed 5,000 project managers in home markets and over 25,000 engineers.

Military Air Solutions (MAS) was one of BAE Systems' UK business units formed during restructuring in 2007 with the responsibility for the design, development, manufacture and support of fixed wing military aircraft. MAS employed over 15,000 people across 24 UK sites and had teams operating all over the world from North America and Europe to South Africa and the Middle East. MAS core customers included the Royal Air Force, Royal Saudi Air Force, US Navy and Indian Air Force. The business had programme responsibility for Eurofighter Typhoon, E3D Sentry, Nimrod, VC10, Hawk UK, Harrier, Tornado, F35 Lightning II (JSF), T45 Goshawk and AV8B. Its Air Mission Support & Services division was also responsible for training. The Autonomous Systems & Future Capability division was also seen as a key area for MAS Company's position as a leader in military aircraft technologies. Although funding for this project was provided by the business unit Military Air Solutions, the project was proposed by the Tactical Data Link research team at the Systems Engineering Innovation Centre (SEIC).

The SEIC was founded in 2003 and aimed to be a centre of excellence for Systems Engineering in all areas of research and technology development, teaching and training. The centre was formed by a joint venture between the East Midlands Development Agency (EMDA), BAE Systems and Loughborough University with the aim of creating a unique, state-of-the-art resource for Systems Engineering in the UK, with global implications for most sectors of industry and academia. This venture involved the co-location of industry and university staff. BAE Systems employees from their Technology and Engineering Services (TES) business unit, part of the Shared Services division, were relocated operate from the venture which was located at Loughborough University. TES is a support business for other BAE Systems business units, providing scientists and engineers who collaborate and work in partnership with academia and innovative organisations. At present, the SEIC is home to leading edge systems engineering projects with involvement from not only Loughborough University and BAE Systems, but also in collaborative work on Systems Engineering themes with others such as the Defence Science and Technology Laboratory (Dstl) and Jaguar Land Rover.

The SEIC provided the support and office environment for this project. The Tactical Data Link team operated from the SEIC, the team was highly independent and distributed, operating from various sites. The team consisted of three core scientists and engineers who have been working on modelling of Tactical Data Links since 2005. The team was not only supplemented by this research project but also involved subcontracting other BAE Systems staff on short term projects for exploitation of research.

1.2.2 THE PROOF OF CONCEPT DOMAIN

To theorise and demonstrate the concepts of hybrid semantic-document models, a suitable domain was needed. The domain chosen was that of Tactical Data Links (TDLs) in military communications. It was an area of modelling work at BAE Systems and the SEIC as well as being a complex engineering domain with much documented material. In particular as TDLs are a family of military standards the work presented in this thesis is derived from the use of Tactical Data Link 16 as its proof of concept domain. BAE Systems has invested development resources into the identification and maturity of class models for the Link 16 message and data structure.

1.2.2.1 Tactical Data Links

The TDL provides one of the backbone technologies underpinning the defence community's goal of Network Enabled Capability (NEC) by providing the information and infrastructure to afford users with both an integrated picture of the battlefield and also provide tasking orders and responses. The TDLs have evolved from NATO's Standardization Agreements (STANAGs), the United States Department of Defense (DoD) then used these STANAGs to derive the Military Standards (MIL-STDs) for TDL. This has resulted in the TDL domain being defined in two families of standards: NATO's STANAGs and the DoD's MIL-STDs.

Due to the nature of joint and combined operations a number of TDLs are in service with coalition forces, and are implemented on a variety of assets, such as aircraft, ships, land vehicles, and command stations, several variations of Tactical Data Links have evolved to interface with specific unit types. As such, the TDL domain is often referred to as a family of standards. The DoD identifies these variations of the TDL (TADIL) domain via a postfix identifier (A, B, C, F, J, and K). These variations may differ in waveforms, bandwidths, protocols and capabilities and, in addition to these, a number of TDLs are under development to support specific roles, such as the control of Unmanned Aerial Vehicles (UAVs) and intelligence video feeds. Figure 1-1 demonstrates these variations and illustrates the TDL domain;

Table 1-1 also shows the cross reference between TDL specifications and the Military Standards and Standardization Agreements.

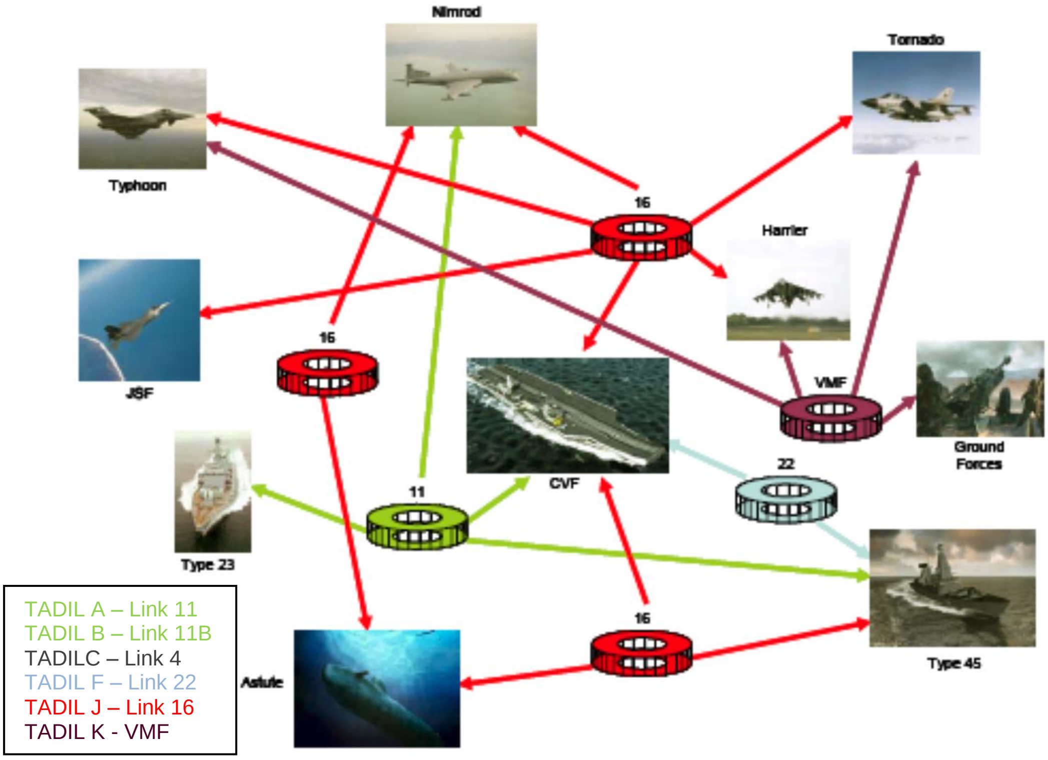


Figure 1-1: TDL Visualisation

Source: (Johnson 2007)

Although it is possible for TDL platforms to operate on multiple TDLs concurrently (e.g. in the example shown in Figure 1-1, the Nimrod aircraft is shown as communicating on both Link 11 and Link 16) acting as a bridge (known as a Forwarding Unit), the domain used in this project was restricted to single Link 16 implementations only. For the purposes of this project it was decided to use the Link 16 or TADIL-J standard. This is described in MIL-STD-6016C (U.S. Department of Defense 2004) and STANAGs 5516 (NATO 2006) and 4175. As both STANAGs are required for the complete representation of Link 16, the MIL-STD version was used as the basis for this project's interpretation.

Table 1-1: Standards supporting TDL

Message Catalogue	Communication Network	Communication Element	Standard	
			MIL-STD	STANAG
TADIL A	Link 11		6011	5511
TADIL B	Link 11B		6011	5511
TADIL C	Link 4A		6004	
TADIL F	Link 22			5522
TADIL J	Link 16	JTIDS, MIDS	6016C	5516, 4175
TADIL K	N/A	VMF, AFAPD	6017	
N/A	Link 14			5514

Source: Johnson et al. 2006

The TDL standards are written to cover the full spectrum of applications. When a TDL is to be implemented on a platform (generically used to refer to the object that the TDL is being integrated into, e.g. ship or aircraft), only the required subset for the platform’s role is implemented. For example, a transport aircraft does not need the fighter and bomber specific messages, so they are not implemented. Note that Link 16 does not define roles and, therefore, the subset selection is determined by the platform implementation team. Due to this, every platform is required to produce additional documents based on the standard. Figure 1-2 shows how the platforms record what they have implemented. These platform documents describe which messages are supported and their behaviour and statements and reasons for non-compliance. In practice, although defined in the standard, the National and Service level document is currently not used widely.

TDL standards are also updated by change documents known as DLCPs. These are documents with step by step revision instructions, e.g. remove X, add Y, replace Z with A etc.

Changes to the standard are rare and projects are usually contracted to deliver against a particular version.

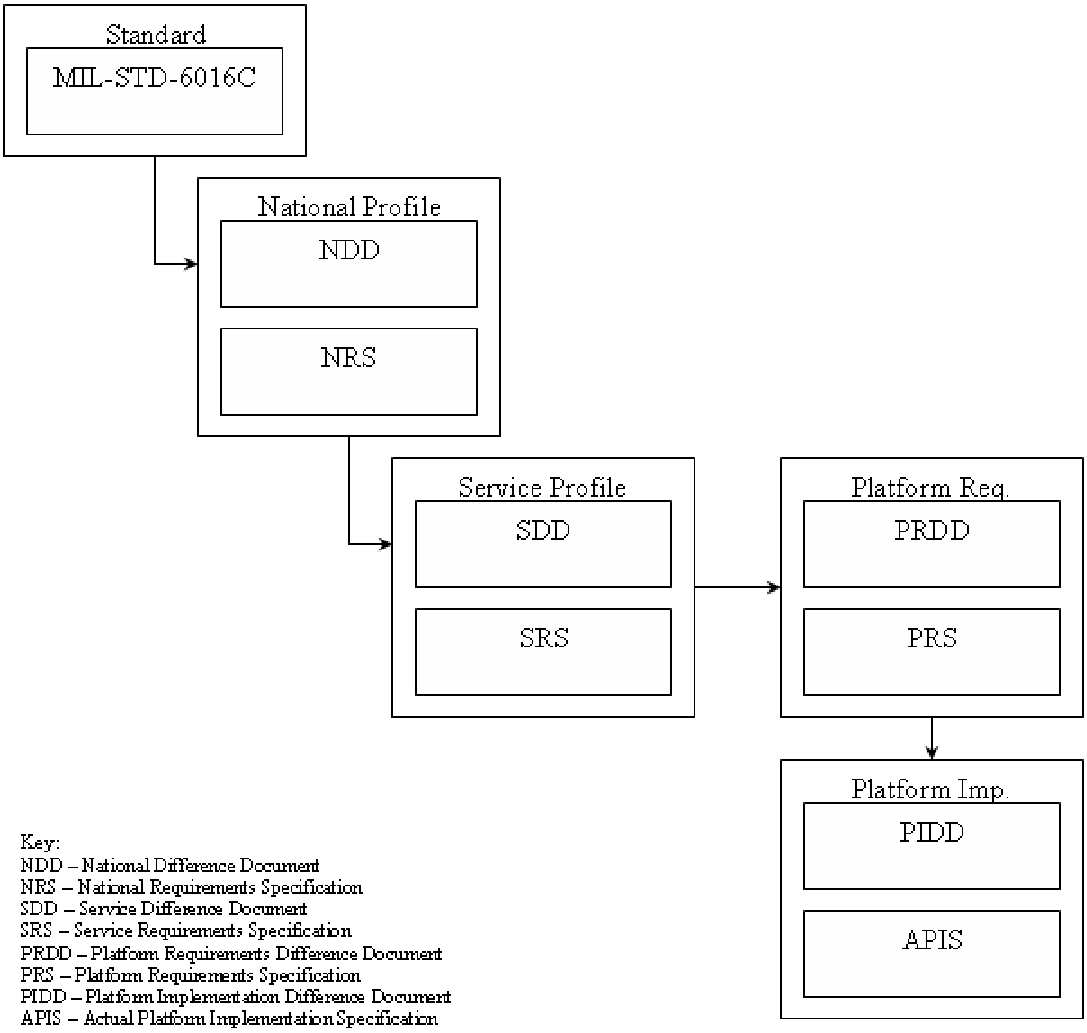

Standard
MIL-STD-6016C
National Profile
NDD
NRS
Service Profile
SDD
SRS
Platform Req.
PRDD
PRS
Platform Imp.
PIDD
APIS
Key:
NDD – National Difference Document
NRS – National Requirements Specification
SDD – Service Difference Document
SRS – Service Requirements Specification
PRDD – Platform Requirements Difference Document
PRS – Platform Requirements Specification
PIDD – Platform Implementation Difference Document
APIS – Actual Platform Implementation Specification

Figure 1-2: Document Hierarchy

Source: (U.S. Department of Defense 2004)

The Link 16 TDL is a general purpose TDL, in contrast to some others, e.g. Link 4A or Variable Message Format (VMF), a list of data link characteristics is provided elsewhere (Holmes & Johnson 2005). It evolved over a number of years, stemming from a requirement identified by the US military in the early 1970s for a TDL offering a broad range of functions that would be applicable for use across multiple forces (e.g. Navy, Marines, Air Force, Army, etc.).

The Link 16 TDL is described in the form of narrative combined with many tables and relatively few figures. At the lowest level of granularity there exists a Data Dictionary identifying the set of types defined for use on the link, these types are identified by a unique key the Data Field Identifier (DFI) and Data Use Identifier (DUI) pair, referred to as the DFI/DUI. The set of messages that may be transmitted over the link are defined in the form of a Message Catalogue. Messages are functionally-oriented and contain a number of words, each of which contains a number of fields, the type of which is defined by reference to the relevant item in the Data Dictionary (the DFI/DUI). Hence, Link 16 messages are tree-structured and must conform to certain well-formed constraints, e.g. all bits in each J-Word (a part of the message) must be associated to a DFI/DUI (i.e. all fields must have a defined type). Such constraints have been captured in the models and are described elsewhere (Holmes et al. 2007). There are a small number of different types of word, and certain elements of the payload are mandated by the word type.

Although platforms typically only utilise their platform specific documentation, this project focussed on the top-level standard to avoid the many security issues that can be associated with the lower-level material. The platform specific documentation is also a specialisation and sub-set of the standard. As such, models and tools accessible to the standard are highly likely to be applicable to the platform specific documents.

1.3 AIMS AND OBJECTIVES

The philosophy of hybrid semantic-document models is to provide a method to facilitate relationships between text-based document views of information and more rigorous computationally accessible representations of its semantic content. Regulatory requirements enforce a document-centric culture, and engineers themselves come from many engineering disciplines and generally consider document based standards or requirements as familiar and comfortable. Although military standards are highly structured and rigorous, they are not in a format accessible to computer processing. Computer aided processing of standards offers several beneficial processes that can enhance the efficiency and productivity of engineers using the standards. However, a solely computer modelled standard becomes inaccessible to the engineers.

Therefore, the aim of this project was to improve the usability of complex engineering documents, in particular military standards documents, by tightly coupling document renderings and prose with a formal machine accessible model of the information in such documents so that engineers can generate the respective views that they are comfortable with (e.g. traditional document based). In reviewing this approach, there are two perspectives to consider when applying the philosophy. Firstly, managing legacy information maintained in a prose/non-model based document and, secondly, the generation of new documents to aid in the consistency and transfer of understanding and knowledge.

1.3.1 OBJECTIVES

Four key objectives were identified to facilitate successfully meeting this aim:

A1. Improve navigation to enable users to have easier information location, search and/or recall.

A2. Help facilitate a user’s understanding of complex information.

A3. Enable improved quality control of material.

A4. Enhance document comparison.

1.3.2 ANTICIPATED ACTIONS AND BENEFITS

At the start of the project it was felt that by achieving the aims and objectives the following actions and corresponding benefits would be achieved:

B1. *Capture the interpretation of complex prose in a more formal mechanism.*

This allows the information author to express their intended action explicitly (removing ambiguity) or where the document already exists, an expert can capture their interpretation which can then be shared by document users to gain a shared consistent knowledge.

B2. *Identify and resolve/prevent inconsistencies within the document.*

Mapping semantic data to its required positions within a document reduces the possibility of errors of inconsistency being introduced through duplicated data.

B3. *Identify and resolve/prevent missing data.*

By modelling the information, cases where references to non-existent data can be captured, resolved and/or avoided. The use of model validation tools would enhance this ability to detect these instances.

B4. *Update information more effectively.*

An update to the semantic model would be reflected in all positions of the document. This would make updating and maintaining the document easier.

B5. *Provide alternative navigation structures.*

It would be possible to provide alternative navigation menus to the traditional tree-like, hierarchical document structure. Similarly it would be possible to restrict the amount of information displayed by restricting the information according to semantic constraints rather than document hierarchies.

B6. *Enhance document information searching.*

It would be possible to enhance traditional document searching through the use of the semantics. Search methods could involve querying the semantic models to return related points within the document.

B7. *Use of model execution facilities.*

By having the core semantic information of the document modelled, it would be possible to utilise model execution facilities to gain computational benefits from the document. This could include code generation, model querying, model validation, etc.

B8. *Perform advanced document comparison.*

It would be possible to compare the semantics of multiple documents to assess their compatibility and/or similarity beyond a simple text comparison available for a traditional document.

1.3.3 BUSINESS REQUIREMENTS AND CONSTRAINTS

As an Engineering Doctorate this project had to fulfil the requirements to deliver research and innovation with a strong business industry-based focus. As such, there were two business requirements and constraints that the project had to work within. These were:

R1. To be able to rapidly prototype and demonstrate functionality in an iterative process to maintain stakeholder support and sponsorship of TDL modelling research.

R2. To utilise only software/tools already approved and authorised.

Maintaining the stakeholder support and sponsorship was critical in this project and the wider context of research into Tactical Data Links at the System Engineering Innovation Centre. Every 12 months, funding was reviewed, and the ability to demonstrate research was needed to exploit and enhance the prospect of maintaining wider research funding. This also meant that the research focus would shift to concentrate effort in areas as directed by funding pressure to provide visibility of research exploitations.

1.4 THESIS STRUCTURE

This thesis documents the research undertaken. This section provides a précis for each of the main chapters of the thesis.

Chapter 1 - Introduction, introduces the research project and its context including the sponsor. It provides background information for a proof concept domain as well as identifying the aims and objectives.

Chapter 2 -Literature Review, provides an assessment of related work that is judged to be suitable to be used in guiding or evaluating the research in working towards achieving the aims and objectives.

Chapter 3 - Research Methodology, defines the scientific process that has been applied to the research to aid the reader in understanding the reasoning for the approaches undertaken in the project.

Chapter 4 - Improving Usability, focuses on analysing the available methods for utilising the case study material and how the usability can be improved to meet the objectives of the project.

Chapter 5 - Improving Storage and Risk Migration, reviews the storage methods available and proposes improved methods to achieve the objective of improving quality by utilising a single data source.

Chapter 6 - Hybrid Models, is a core concept for the project. The chapter proposes a framework to enable elements in the document model and semantic model to be related thereby realising the main aim of the project.

Chapter 7 - Templating, examines the proposed enhancement of templating which targets automated repetition to reduce the probability of inconsistency and missing elements.

Chapter 8 - Summary and Conclusions, collates and reviews all proposals against the projects aims and objectives to constructively evaluate the successes of the project and its contributions to knowledge. It also includes suggestions on areas to extend the research further.

Figure 1-3 provides an overview of the chronology events performed by involved parties with reference to the chapter where the event is discussed.

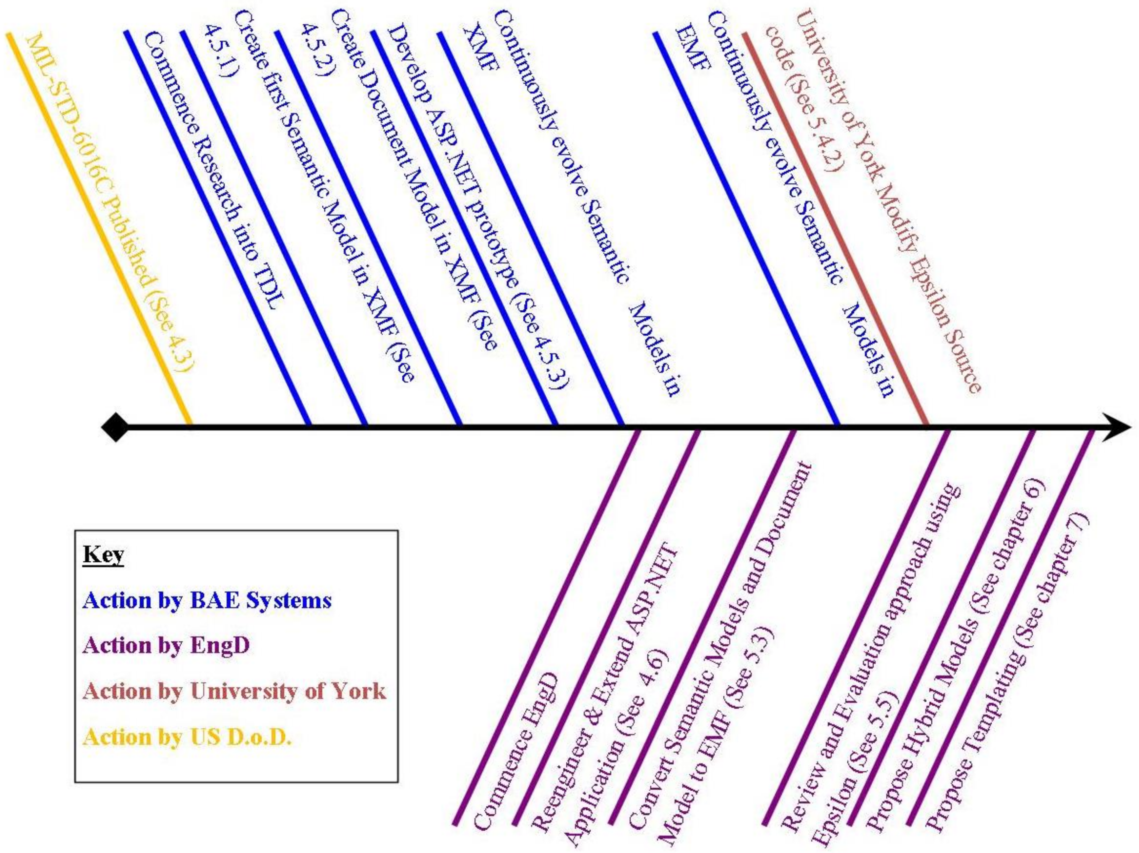

MIL-STD-6016C Published (See 4.3)
Commence Research into TDL
Create first Semantic Model in XMF (See 4.5.1)
Create Document Model in XMF (See 4.5.2)
Develop ASP.NET prototype (See 4.5.3)
Continuously evolve Semantic Models in XMF
Continuously evolve Semantic Models in EMF
University of York Modify Epsilon Source code (See 5.4.2)
Commence EngD
Reengineer & Extend ASP.NET Application (See 4.6)
Convert Semantic Models and Document Model to EMF (See 5.3)
Review and Evaluation approach using Epsilon (See 5.5)
Propose Hybrid Models (See chapter 6)
Propose Templating (See chapter 7)
Key
Action by BAE Systems
Action by EngD
Action by University of York
Action by US D.o.D.

Figure 1-3: Thesis Chronology

2 LITERATURE REVIEW

CHAPTER PREFACE

This chapter provides a literature review of several related topic areas that have impacted on this research. They are provided to give context and demonstrate where the approaches undertaken in this research differ from that already published. It includes sections from the literature discussed in all papers published by the author during this research, including a significant portion of paper 5 (See Appendix E).

2.1 INTRODUCTION

Before undertaking any research project it is important to critically assess what has been conducted and published in the field(s) already. This chapter documents those reviews to provide the reader with background knowledge and the context of related work which has guided this project. It includes several sections which were relevant during the undertaking of this project. Primarily, it was important to assess if any work already achieved the aims, hence section 2.2 - Combining Formal Models with a Document.

A core aim of the project was to combine information captured in a document with its semantic representation. As such, a method for modelling this document structure was required. Section 2.3 - Document Meta-Models, reviews these methods. As the project evolved, the methods for persisting data needed to be reviewed. As the models were complex, an understanding of persistence technology was required to review the suitability of approaches, which is in section 2.4 - Persistence Methods.

The domain and chosen case study of TDL Link 16 is a huge document and any improvement in usability and navigation was only going to be achieved through electronic representation. Section 2.5 - Electronic Document Rendering, reviews electronic rendering and navigation techniques to understand how the MIL-STD-6016C can be improved. Due to the business constraints R1 (Utilise already approved tools) and R2 (Rapid exploitation technologies), web based exploitation was going to be a major technology for providing proof of concept. As such, it was deemed necessary to understand the performance and user expectations that any exploitation must adhere to, which are provided in section 2.6 - Web based Performance. As the intention was to utilise models and web exploitation, it was believed necessary to review the field of Model Driven Web Engineering (MDWE) to understand if any benefit could be obtained through this technology. The review of MDWE is found in section 2.7 - Model Driven Engineering.

2.2 COMBINING FORMAL MODELS WITH A DOCUMENT

There appears to be little research in combining a formal model with document markup. Much investigation has been carried out into defining models and/or ontologies to add meaning to complex documents (Guo et al. 2004, Palmirani et al. 2009), but this does not help with the execution or other model management functions such as code generation or model validation. By using predefined domain markup, more generalised queries can be applied to the document, which would only move part way to the overall aim. It is predominantly a technique to assist with information retrieval tasks, for which much has been discussed on how to model the document structure to aid the retrieval process. This too, does not help in achieving the aim.

Similarly much research has been conducted into requirements traceability (Antoniol et al. 2002, Marcus & Maletic 2003, Ramesh & Jarke 2001, Spanoudakis et al. 2004). Although the MIL-STD could be considered as a requirements document, the research on requirements traceability is focused on tracing between chains of documents. Even if the semantic model is treated as an XML file to facilitate requirements traceability techniques, it still would not help reduce the duplication of data between the two models and would make utilising the models more difficult.

One of closest examples of related work is by Winter and Nittel (Winter) who discuss the geospatial standard and address the need for more formal modelling. Although they do stress the relationship between prose and the models, there is no direct traceability and the information is duplicated. They make use of functional programing languages which, given the wide range of engineers anticipated to use this solution, would be likely to be confusing.

During workshops with a cross section of engineers using the MIL-STD, it was found that even UML style modelling was not widely understood. However, the closest related area of research appears to be the Health Level 7 Clinical Document Architecture (CDA) framework (Dolin & Alschuler 2011, Dolin et al. 2001, Dolin et al. 1999, Dolin et al. 2006).

2.2.1 HEALTH LEVEL 7 CLINICAL DOCUMENT ARCHITECTURE (CDA) FRAMEWORK

The framework has two core parts, the Clinical Document Architecture (CDA) and the Reference Information Model (RIM). This is equivalent to the proposed approach for complex engineering domains, a document model similar to the CDA and a semantic model representing the RIM. The aims of the framework are not too dissimilar to this research:

> “The need for a clinical document standard stems from the desire to unlock the considerable clinical content currently stored in free-text clinical notes and to enable comparison of content from documents created on information systems of widely varying characteristics. Given the variability in clinical notes, including structure, underlying information models, degree of semantic encoding, use of standard healthcare terminologies, and platform- and vendor-specific features, it is currently difficult to store and exchange documents with retention of standardised semantics over both time and distance.” (Dolin et al. 2001)

The RIM provides a vocabulary (ontology) and the meta-model to construct semantic information within a document. This inclusion of semantic information in the document is a primary aim for this research. The CDA is a document markup standard that specifies the structure and semantics of clinical documents. Explicit markup tags are defined that facilitate the encoding of text according to the ontology. The text is not directly retrieved from the underlying RIM, the ontology is used purely to encode the text string. In addition to utilising the ontology, the CDA also allows for the recording of process and interactions through the use of instances of the RIM model. These instances of the RIM require the reuse of the data recorded in the textual elements leading to duplication. There is no attempt made to reutilise the information recorded in either part, although CDA does record both the instances of both models within the one document.

There is criticism of the RIM (Hasman et al. 2006, Schadow & Mead 2006), however, as the RIM itself provides no benefit to the thesis domain as it is an information model for clinical workflows, this criticism can be ignored. The CDA and RIM at present have also deliberately avoided advanced and complex semantics such that a staged approach to the implementation and delivery of a shared structure and semantics can be achieved (Dolin et al. 2001). This too is something that is considered suitable for complex engineering domains by growing and evolving the semantic model over a period of time allowing for user adoption and providing time to establish the complex semantics.

An astute feature provided with the CDA is the ability to locally define markup (Dolin et al. 2006) which can extend the CDA semantics provided that these extensions do not change the meaning of any of the standard data items, and can be safely ignored by applications without the local knowledge.

The CDA approaches the problem where there are multiple sources of information that need to be standardised to enhance interoperability. The project described in this thesis works in the opposite direction. The source is singular and well controlled, however it requires better methods to disseminate the information to different users with different requirements.

2.2.2 CONCLUSIONS

There appears to be little research in combining a formal model with document markup. The closest example of this is the Health Level 7 Clinical Document Architecture (CDA) framework. The aims of the framework were not too dissimilar to this research either. However, the main difference between the projects is that, in the MIL-STD-6016C project, the source is singular and well controlled and the project requires better methods to disseminate the information to different users, whereas the Health Level 7 project aimed to solve the opposite problem.

2.3 DOCUMENT META-MODELS

A core aim of the project was to combine information captured in a document with its semantic representation. As such, a method for modelling this document structure was required. There has been much research (Brugger et al. 1996, Jain & Yu 1998, Klink & Kieninger 2001) conducted into modelling documents for the purpose of information retrieval or for structure recognition of scanned documents. However, within this research, the term document model was aimed at describing and modelling the structure in terms of heading, paragraph, table, etc. This more closely aligns to the field and concept of document markup.

There are lots of document markup languages available. Several were considered unsuitable for this project which included those designed for the markup of comments within programming source code such as Javadoc (Oracle 1993) or reStructuredText (reStructuredText 2010), and those of proprietary nature such as Adobe's Maker Interchange Format (MIF) (Adobe 2008) and the Computable Document Format (CDF) (Wolfram 2013). Some of the major languages suitable for complex engineering domains are described in more detail in the next sections.

2.3.1 GENERIC MARKUP LANGUAGES

There are many generic document markup languages which could be utilised for encoding complex engineering domain documents such as HTML and Office Open XML (OOXML). In truth they are all based on XML which too can be utilised as a document markup language. HTML and OOXML provide the ability to encode text into paragraphs, lists and tables, and facilitate the ability to style and include graphics.

2.3.2 DARWIN INFORMATION TYPING ARCHITECTURE (DITA)

The Darwin Information Typing Architecture (DITA) (OASIS 2010) is XML-based. It is maintained by the Organization for the Advancement of Structured Information Standards (OASIS) and was designed for authoring, producing, and delivering technical information, which seems suitable for the concept of complex engineering domains. It has a topic-oriented information architecture which does not really fit with the style of the MIL-STD-6016C. It is, however, an extensible architecture which suits the proposal of hybrid semantic-document models as it would be possible to extend this model. It allows for the automatic generation of documents in many different formats including XHTML, Compressed HTML Help (.chm), PDF, Eclipse Help, JavaHelp and Rich Text Format (RTF). An example of DITA markup is provided in Figure 2-1.

```
<task id=”MakingTomatoSandwich”>
   <title>Making a tomato sandwich</title>
   <taskbody>
      <prereq>To make a tomato sandwich, you will need:
         <ul>
            <li>a sharp knife</li>
            <li>a cutting board</li>
            <li>one or more fresh tomato</li>
            <li>two slices of bread</li>
            <li>mayonnaise</li>
         </ul>
      </prereq>
      <steps>
         <step><cmd>Using the knife, spread the mayonnaise on one side of each
slice of bread.</cmd></step>
         <step><cmd>Place the slices of bread, mayonnaise side up and side by
side, on one end of the cutting board.</cmd></step>
          <step><cmd>Using the knife and the other end of the cutting board, cut
the tomato or tomatoes into 1/4” slices.</cmd></step>
         <step><cmd>Place the tomato slices on one slice of bread.</cmd></step>
         <step><cmd>Place the other slice of bread, mayonnaise side down, on top
of the tomato slices.</cmd></step>
      </steps>
      <result></result>
      <postreq>Eat the sandwich before bread gets soggy</postreq>
   </taskbody>
</task>
```

Figure 2-1: Example DITA Markup

Source: (Benz 2010)

2.3.3 DOCBOOK

DocBook (OASIS 2009) has existed since 1992. It too is now maintained byOASIS. It is a schema that is published in several different languages including XML DTD. A key aim for DocBook was the support for books and papers about computer hardware and software. DocBook enables its authors to create document content in a presentation-neutral form and subsequently generate an output format. Many different formats are supported including

HTML, XHTML, EPUB, PDF, and HTML Help. An example of the DocBook markup is provided in Figure 2-2.

```
<article xmlns='http://docbook.org/ns/docbook'>
 <title>Example example</title>
 <example xml:id="ex.dssslfunction">
 <title>A DSSSL Function</title>
 <programlisting>
 (define (node-list-filter-by-gi nodelist gilist)
   ;; Returns the node-list that contains every element of the original
   ;; nodelist whose gi is in gilist
   (let loop ((result (empty-node-list)) (nl nodelist))
       (if (node-list-empty? nl)
      result
      (if (member (gi (node-list-first nl)) gilist)
          (loop (node-list result (node-list-first nl))
               (node-list-rest nl))
          (loop result (node-list-rest nl))))))
    </programlisting>
 </example>
</article>
```

Figure 2-2: Example DocBook Markup

Source (Walsh 2010)

2.3.4 TEXT ENCODING INITIATIVE (TEI)

The Text Encoding Initiative (TEI) (TEI Consortium 2013) is produced by the TEI consortium, it can be considered similar to DocBook and DITA in that an ontology exists that can be used to encode text within a document. It was primarily developed to support documents in the humanities, social sciences and linguistics. An example of the markup can be seen in Figure 2-3.

```
<TEI xmlns="http://www.tei-c.org/ns/1.0">
 <teiHeader>
  <fileDesc/>
  <profileDesc>
   <particDesc>
    <listPerson>
     <person xml:id="AH">
      <p>Alan Hodge. Oxford history graduate. Became close friends with Laura
Riding &amp; Robert Graves. First husband of Beryl Graves.</p>
     </person>
     <person xml:id="BP">
      <p>Beryl Pritchard. Daughter of Harry and Amy Pritchard, Robert Graves's
second wife. Formerly married to Alan Hodge. Robert and Beryl had four children:
William, Lucia, Juan and Tomas.</p>
     </person>
     <person xml:id="DR">
      <p>David Reeves. Brother of James Reeves.</p>
     </person>
    </listPerson>
   </particDesc>
  </profileDesc>
 </teiHeader>
 <facsimile>..</facsimile>
 <text>
  <front>...</front>
  <body>
   <div type="diaryentry" n="1938-10-10" facs="#graves1938-10-10-1">
    <head>Oct 10
    <del>Tuesday.</del>
     <add>Monday</add>
    </head>
    <p>Ghost, completing <abbr>ch</abbr> IX</p>
    <p>Dictionary with <rs ref="#AH">Alan</rs>.</p>
    <p>A lot of time goes to making charcoal for 'Marthe',
<rs ref="#BP">Beryl</rs>'s now using this
    <foreign>fugon</foreign> <note>charcoal-burner</note> for warming her
attic.</p>
    <p>Went to Montauban with <rs ref="#DR">David</rs> – first visit for about ...
    </p>
   </div>
  </body>
 </text>
</TEI>
```

Figure 2-3: Example TEI Markup

Source: (Rahtz & Cummings 2013)

2.3.5 CONCLUSIONS

There are many different methods for the markup of documents ranging from the simplistic such as HTML to those incorporating an element of semantic markup such as DITA and TEI. No particular solution is best suited to the capture of TDL Link 16 document. Those that include semantic markup simply add a level of complexity to the document model that could detract from the project's aims by over complicating the model in any proof of concept. As all methods, in truth, are an extension of XML, a simplified document model based around XML would more easily enable the business constraint of rapid exploitation (R1) to be achieved. On completion of proof of concept, future work should enable the change to support more complex standard document models.

2.4 PERSISTENCE METHODS

This project inherited an application from BAE Systems (See section 4.5.3) which utilised a Relational Database to store models defined in an object-oriented manner. This section reviews the literature assessing the suitability of the persistence method to discover if a better alternative existed for the project. The storage method needed to be able to support multiple inheritance objects.

There, typically, have been two main types of database management system, Relational Database Management Systems (RDBMS) and Object Database Management Systems (ODBMS) or Object-Oriented Database Management Systems (OODBMS) (Joan et al. 2011). The main comparison can be seen in Table 2-1.

Table 2-1: Comparison RDBMS Vs. ODBMS

ODBMS	Advantages over RDBMS	Disadvantages over RDBMS
1	Easier Navigation	Lower efficiency when the data and its relationships are simple
2	Better concurrency control	Relational tables are simpler
3	Less code required when applications are object-oriented	More user tools exist for RDBMS
4	Data model is based on the real world	Standards for RDBMS are more stable
5	Works well for distributed architectures	Late binding may slow access speed
6	Reduced paging	Support for RDBMS is more certain and changes are less likely to be required

Source: (Joan et al. 2011)

This difference between RDBMS and ODBMS is referred to as the object-relational impedance mismatch problem (Ambler 2003). Object Relational Database Management Systems (ORDBMS) are one approach used to tackle this problem. ORDBMS are a hybrid between RDBMS and ODBMS where the RDBMS has been extended to include an object model that supports more complex types such as arrays and structures. An ORDBMS supports several of the complex requirements of an ODBMS by providing complex types whilst, at the same time, benefiting from the RDBMS features such as speed, querying and scalability. There has been much reviewed in the comparison of the persistence techniques (Kalantari & Bryant 2012, Stonebraker & Moore 1995, Vanathi & Uthariaraj 2011). The general opinion is shown in Figure 2-4.

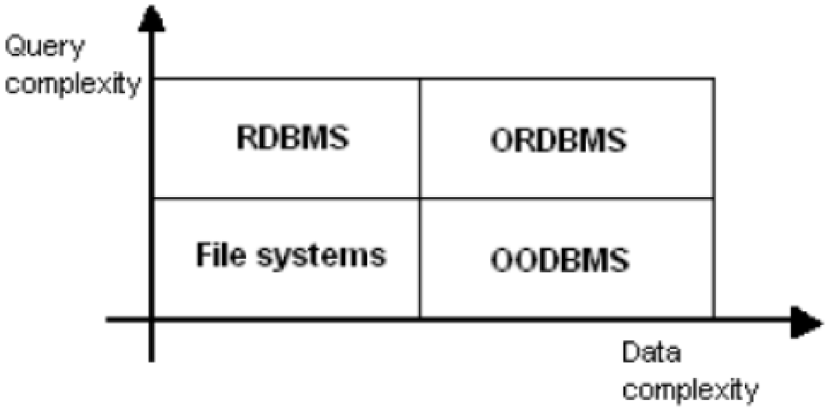


Figure 2-4: Choosing Persistence Method

Source: (Stonebraker & Moore 1995)

An alternative solution to the object-relational impedance mismatch problem is to utilise an Object-Relational Mapping (ORM) framework (Ogheneovo et al. 2013). An ORM can be employed to automatically manage the mapping from application objects to RDBMS or ORDBMS objects. This abstracts the complexity of object mapping away from the developer allowing for object-oriented development with RDBMS storage.

2.4.1 CONCLUSIONS

Persistence is a well-researched domain. This research project requires a high degree of querying and contains complex data. Given this, literature is very clear that the most suitable method for persistence is an ORDBMS. However, a suitable alternative would be to utilise an Object-Relational Mapping (ORM) framework as this would remove the need to actively consider the persistence method and remove any need for a manual generation of suitable database schemas.

2.5 ELECTRONIC DOCUMENT RENDERING

A review of literature to understand the recommended practices for the rendering of electronic documents was performed to enable the project to improve the current representation methods. As the case study document (MIL-STD-6016C) is so large, understanding the best practices to avoid disorientation and allow for improved usability through better recall and understanding were of critical importance.

2.5.1 ELECTRONIC REPRESENTATION

The medium used is believed to affect the comprehensibility of texts. It is commonly believed that reading from paper is both easier and more effective than reading from a computer screen. There has been much research in paper versus screen presentation of texts (Dillon 1992, Gulbrandsen et al. 2002, Muter & Maurutto 1991, O'Hara & Sellen 1997, Piolat et al. 1997, Ventura 1988). Most look at assessing the reasons behind the differences in comprehension with Dillon (Dillon) giving a comprehensive critical review of the field. Ventura (Ventura) however, considers the environmental and logistics of large technical manuals. She argues that it is impractical to use paper based versions of large documents, given their weight and size. This too is true for the TDL Link 16 with its 7300 pages being equivalent to approximately 15 reams of paper, which is enough to fill two entire filing cabinet drawers.

The comprehension studies are commonly based around Schumacher and Waller's (Schumacher) theory of outcome and process measures for usability. They spilt groups of measures into these two groups. Outcome measures are those based on the result of reading, such as speed and accuracy of recall. Process measures are those focused on the function of reading such as eye movement and navigation. Although many studies have been conducted, it is difficult to draw any solid conclusion from earlier literature as often the experiments suffered from poor experimental design and were of too narrow scope (Dillon 1992, O'Hara & Sellen 1997). The experiments often drew on only a small sample size between 10 and 25 people from which the conclusions were drawn.

With the primary objective of this research being the improvement in usability of the TDL standard, outcome measures for speed, accuracy and comprehension are of most interest, with process measures such as eye movement (Dillon 1992, Schumacher & Waller 1985) being beyond scope. However, the navigational process issue and spatial considerations are of possible critical importance given the size of the document.

Previous research by Muter et al (Muter) had found that screen reading was 28.5% slower than reading from paper, as did much of the literature reviewed by Dillon (Dillon). However, Muter and Maurutto (Muter) revisited their research to conclude that, given advances in the technology of screen quality, reading from a screen was comparable to reading from paper for both speed and comprehension. Although Dillon draws the conclusion that reading from a screen is slower, his conclusion is based on older material while Muter and Maurutto come to a more up to date conclusion that, given the improved screen quality, reading from screen can be comparable to that of reading from paper

Annotation or note-taking while reading is believed to be a major aid in comprehension (Guimbretière 2003, O’Hara & Sellen 1997). This is primarily a benefit to those aiming to learn from the document, in particular the O’Hara and Sellen (O’Hara) experiment asked readers to write a summary after reading, which led the reader into taking notes. Although users of the electronic text were allowed to take notes in another document, users felt uncomfortable and dissatisfied. Given the advanced nature of the TDL environment and sheer size of the document this is not considered a contributing factor for comprehension in this domain. However, research has been conducted on developing systems that would allow the user to make freehand annotations to electronic documents as they read (Guimbretière 2003, Price et al. 1998, Schilit et al. 1998, Wellner 1993).

The spatial layout of text has an effect on the process of reading, including the spatial layout in the environment and the spatial layout of the text on a page. The spatial layout assists readers to recall or search for information. Readers often create a mental picture of the document such that the location of information can be recalled in relation to their mental picture (Ahmed & Blustein 2005, Dillon 1992, Gwizdka & Spence 2005, O’Hara & Sellen 1997, Piolat et al. 1997). O’Hara and Sellen (O’Hara) found that readers used their sense of touch to acquire a sense of position within the document which is used to make the mental image. Piolat et al. (Piolat) also found that having the page number visible also helped build this positional opinion. Both Dillon (Dillon) and O’Hara and Sellen (O’Hara), found that this positional information was harder for electronic versions as a full page view is rarely achieved given screen sizes. Dillon does, however, record that performance can be improved through larger screens.

Because of this space limitation, most electronic documents employ scrolling. Dillon (Dillon) found that novice users found comprehension harder when scrolling compared to paging, but concluded no difference for others. Piolat et al. (Piolat) found that scrolling restricted a user’s ability to form a mental picture and a paging technique was much better for locating recalled information. They also found that although paging required more movement and actions, its users did not read the document any more slowly. O’Hara and Sellen (O’Hara) discovered this movement can actually aid reading, the ability to quickly move to a cross reference page helps. This finding supported Dillon’s opinions that a key benefit to electronic texts is the possible use of multiple windows, which a user can rapidly switch between.

In contrast to those who conclude readers gained a better understanding by using a paging layout rather than scrolling, Gulbrandsen et al. (Gulbrandsen) found that users appeared to find reading PDF documents more difficult than paper, whereas their HTML readers did not experience this difficultly. This would suggest that users performed better with scrolling. Unfortunately the study is unable to make this conclusion as it primarily investigated the differences between paper and screen reading in different languages. It was only the Danish experiment that used PDF, the Norwegian and Swedish experiments used HTML. Similarly the screen size was not kept consistent so this too could have had an effect. This highlights Dillon’s (Dillon) and O’Hara and Sellen’s (O’Hara) comments that some experiments lack the necessary control over experimental design parameters. There appears to be little similar research, suggesting that particular applications can affect the screen reading ability. Nielsen (Nielsen) believes that the use of PDFs in a web-based solution is a crime against usability. In 2003, he listed several reasons for not using PDF on screen, which he still believed true in 2008 (Nielsen 2007).

2.5.2 Document Navigation Considerations

Research into navigation can be segregated into two main categories, cognitive overhead and disorientation. Both topics are closely related in that the navigation structures affect the cognitive load and cognitive style will affect the degree of disorientation.

Studies show that the cognitive style of a reader affects their navigation (Calcaterra et al. 2005, Chen 2002, Chen & Macredie 2002, Dufresne & Turcotte 1997, Mitchell et al. 2004, Park & Kim 2000). They focus on the perspective of the reader as a learner and spilt users into field dependent and field independent learners. They can be characterised as in Table 2-2.

Table 2-2: Field Dependent and Independent Characteristics

Field Independent		**Field Dependent**	
Characteristic	*Preference*	*Characteristic*	*Preference*
Active Approach	Prefer to use index to locate specific items	Passive Approach	Rely on map to impose structure
Analytical Tendency	Prefer depth-first paths	Global Tendency	Prefer breadth-first paths
Internally Directed	Prefer non-linear and flexible navigation	Externally Directed	Prefer linear and restricted navigation

Source: (Mitchell et al. 2004)

The common finding across all the studies is that the navigational process should be tailored to the reader's cognitive style (Chen 2002, Chen & Macredie 2002). Independent learners are believed to favour a non-linear navigation with the freedom to explore on their own. In contrast, dependent learners require a linear structure relying on the designers to propose the intended path (Chen & Macredie 2002). There is little published evidence on whether these cognitive styles continue to be effective in other document uses, such as reference.

Interface tailoring on cognitive style has led to the development of Adaptive Hypermedia (Chen 2002, Chen & Macredie 2002, Dufresne & Turcotte 1997, Mitchell et al. 2004). This is where the interface is customised for the user given three main factors:-

1. Directed Guidance, where the link order is modified. For example, Field Dependent learners prefer a breadth first approach whereas independents prefer a depth first.
2. Link Disabling, as dependent learners are easily disoriented and distracted by non-linear links, irrelevant links are either enabled or disabled.
3. Layout, where the structure is varied, e.g. a hierarchical map for dependent learners.

An interesting suggestion is the use of annotated links (Chen 2002) such that the user can gain more understanding of the link before navigating away. However, Mitchell et al. (Mitchell) found no gain for tailored approaches compared to a well-designed interface for all users, but still 75% of both dependent and independent learners preferred having a selection of navigation tools available.

Similarly to cognition, the domain context can affect navigation (Mitchell et al. 2004, Rouet & Bigot 2007, Vaughan & Dillon 2006). Prior knowledge affects a reader's preconception, resulting in an expected navigation structure. Most recently, Vaughan and Dillon's (Vaughan) work has termed this "genre". The results suggested genre does play an important role. The genre conforming versions of a site enabled users to recall the structure of the site better, and this should aid the spatial location of text. It is also believed by Vaughan and Dillon (Vaughan) that it is advantageous to allow users to generate a better mental representation of the site. It is important to note that non-genre-conforming site did improve over time, however, it was not able to match the performance of a genre conforming site.

Nielsen (Nielsen) believed that disorientation is a major problem for usability of hypertext documents. In general, the problem with disorientation is the user's ability to get lost (Elm & Woods 1985, Marchionini 1988). It is generally considered that the problems revolve around users knowing where to go next, where they are, and how they got there. Park and Kim (Park) put these factors down to two groups of contextual information, temporal and spatial. They suggest that the navigation system should aid users in re-establishing their sense of location through the spatial and temporal context. Hierarchical linking appears to be the suggested form of spatial information with a breadcrumb trail acting as a temporal provider.

Disorientation is mostly related to the type of navigation link aid used (Elm & Woods 1985). The main types of aids are termed:

- hierarchical,
- content lists such as an alphabetical index,
- linear such as next and back,
- non-linear such as linking to other sections that are not neighbouring.

Research finds that users get disoriented most when the navigation is focused on non-linear linking (McKnight et al. 1990, McDonald & Stevenson 1996, McDonald & Stevenson 1998, McDonald & Stevenson 1999). In fact, linear linking is considered most efficient for both performance of recall actions and limiting a user's disorientation. Hierarchical and contents list linking is considered to sit between the extremes of linear and non-linear linking. Most experiments seem artificial as the system only uses one navigation aid which is not common in reality. Researchers are able to conclude that there are some design considerations that limit disorientation, such as the use of page numbers and chapter headings. They are found to help a user in their recall of important information and aid the perception of size of the document.

An unusual navigation aid is the use of spatial maps (Calcaterra et al. 2005, McDonald & Stevenson 1998, Nielsen 1990). They do not appear to be utilised widely in the World Wide Web, despite being considered good at lowering disorientation. McDonald and Stevenson's (McDonald) study interestingly found that this technique removed any benefit a user may have had from prior knowledge. Novice and expert domain users both achieved very similar levels of accuracy in recall and speed of navigation. It is not clear whether maps would be suitable for large documents as they could become cluttered, all currently reported research has been conducted with relatively similar-sized maps with a low number of nodes.

Other considerations for navigation include the technique for moving between pages. Research suggests that variable zooming in and out dependent on the speed of scroll best mimics the scanning technique employed when reading from paper (Cockburn et al. 2006, Johnson, J., Holmes, C. & Raiz, A. (2006), ’product standards - tactical data links 2006 final report, Technical Report SEIC-RP-0551, System Engineering Innovation Centre (BAE Systems).

, Sun & Guimbretière 2005).

Weinreich et al (Weinreich) carried out a long term study over 105 days. Their view of navigation considered the web rather than a document and analysed the statistics of clicks such as back, links and submit clicks. Most interestingly for this study were their within-page navigation findings. They found only 0.4% users would scroll horizontally to select a link. Key to successful rapid navigation was the ability of the user to quickly scan the information with limited scrolling. For this purpose they produced a screen pixel real estate chart (Figure 2-5). They believed that design should, as of 2006, cater for 1024x768 pixel resolutions and make the most use of high value pixels (top left of a screen, see Figure 2-5) for navigation. Other design factors were found by Nielsen (Nielsen). He found that subtle changes in background colour, to differentiate sections are not noticed, as a screen full of text does not allow a user to gain the spatial information required to recall the position of this text.

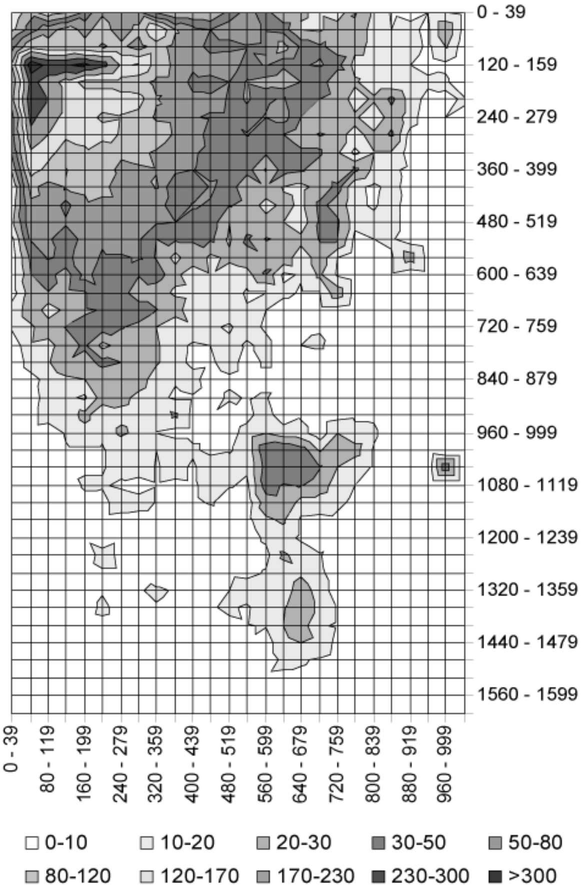


Figure 2-5: Pixel Real Estate Value

Source: (Weinreich et al. 2006)

2.5.3 CONCLUSIONS

Electronic versions of documents are no longer limited by poor resolution and other screen limitations. Although paper has benefits for the reader, for larger documents an electronic version has outweighing benefits, such as search functions and portability. There are two main areas of research in document navigation, cognition and disorientation. Research in cognition has identified user attributes, indicating different reader capability, defined as Field Dependent and Field Independent. Study suggests that an interface should be well designed for all users or be customisable to suit the cognitive style. Disorientation is mainly influenced by the navigational aid being used, linear navigation is found to achieve the best performance with non-linear linking the worst. Studies also suggest that prior knowledge of the domain aids a user in their navigation, although the use of navigation maps helps novice users achieve similar results to expert users. However, some key questions remain unanswered from current literature:

1. Do the conclusions of all this literature still apply? The problem with hypertext is that the Internet is still growing and growing at a remarkable speed. In fact, in 1995 it was estimated that there were 16 million users of the Internet rising to 361 million in 2000 (Miniwatts Marketing Group 2009). That is over a 2000% increase, or 450% increase a year if this is compared with the December 2007 figures of 1.3 billion. The primary target user population for this project are likely to be Europe and North America who, when combined, account for 44% or 572 million users (Internetworldstats 2008). The penetration rate shows similar rapid advancement. Internetworldstats define this as the percentage of the total population that has the basic knowledge of and available access to the Internet. In 2007, North American had 71% and Europe 43% penetration. Within Europe, if only the European Union is considered, then this rises to 55% and, if only the UK, then it is 66% (Internetworldstats 2008). Again, if this is compared to the UK penetration rate in 2000 (30%) it can be seen that this has almost doubled in seven years. All this suggests that Internet usage is now more widely accepted, so possibly the elements of disorientation may no longer apply given the wider and more common place usage of the Internet will enable readers to become used to the non-linear linking style of navigation.

2. Most literature, when investigating navigation, is focused on the user as a learner. Although some studies accept that prior knowledge of the domain aids navigation, they still investigate the process as a learning activity. For the TDL domain the majority of use is for reference or a goal-directed information seeking context, whereby the user already knows a great deal and needs only to locate specific bits of information. This could be classified as the recall action, but in reality the user is not going to read all 7300 pages. It is unclear whether this change in intended goal will produce different findings to that of the research reported in the current literature. There has been some limited investigation into goal-directed information seeking, such as by Gwizdka and Spence (Gwizdka), but no authors address disorientation and few address navigation.

Understandably, the published research investigates the hypotheses on small documents. No research has attempted to investigate whether there is a scaling effect in relation to document size. It is assumed that different navigation techniques will have different effects given the varying size of document. Furthermore, no reported research has investigated if the disorientation problem is multiplied with larger documents.

2.6 WEB BASED PERFORMANCE

A review of web based performance and users' expectations enabled the assessment of whether any application would be useable to target users.

2.6.1 THE INTERNET

The use of Internet-based technologies such as HTML has exploded in the last decade. In 1995 there were approximately 16 million users worldwide (Miniwatts Marketing Group 2009), with approximately 2600 domains (Internet Systems Consortium 2009). In 2005 there were an estimated 1 billion worldwide users (Miniwatts Marketing Group 2009) and 350 million active domains (Internet Systems Consortium 2009). In 2008 the number of Internet users reached over 1.5 billion (Miniwatts Marketing Group 2009) with active domains reaching over 570 million (Internet Systems Consortium 2009).

Similarly the development of new technologies to support this exponential increase in users and domains has made rapid progress. The first commercial modem was released in 1962 and could achieve a speed of 300bps (Dunn 1997). It was not until the 1990s that the more familiar speeds of 14.4kbps-56kbps telephone line modems were developed, with 56kps arriving in 1996 (Dunn 1997). It took until the turn of the millennium for significant uptake of broadband services to be adopted in the UK. Broadband offered faster speeds in excess of 512kbps. Now some broadband suppliers are offering speeds of up to 50Mbps, a 100 fold increase in approximately 10 years (cable.co.uk 2009).

Not only is the number of users and the speed of the underlying technology increasing but so too is the size of web pages (King 2008). King's study concluded that the average size of a web page has grown from 94k in 2003 to 312k in 2008, a 22 times increase on the average size in 1995. However, because the increase in speed from 1995 to 2008 more than matches the increase in file size, King's belief that broadband users now expect faster response times seems sensible. King's study estimates that broadband users' average download time is now 2 seconds, down from 3 seconds in 2003.

2.6.2 WAIT TIMES

The review of advances in the Internet has shown that users now expect fast download speeds. A web-based solution must conform to these expectations otherwise users will consider the solution to be of poor quality due to excessive waiting times. Neilson was one of the earliest researchers to investigate the effect of waiting times of page loads in the mid 90s (Nielsen 1996, Nielsen 2004). Nah (Nah) provided a comprehensive review into tolerable waiting times. Her review found that users expect a computer response to occur within 0.1-10 seconds; however for web-pages this perspective has shifted. She found users were prepared to wait much longer, between 8-41 seconds. There is some supporting research that suggests that 30 seconds is an upper limit (Nah 2004, Rose et al. 2005, Selvidge et al. 2002). However, most other studies show that a web user is willing to accept an 8-10 second delay in loading a page (Nah 2004, Nielsen 1996, Palmer 2002).

Nielsen listed download speed as one of the top ten problems of web usability (Nielsen 1996). In Neilson (Nielsen) it estimated that wait time would still remain a problem until 2011 because of technology, however in 2007 (Nielsen 2007) waiting times no longer made his list of top ten problems for usability.

Most research suggests that having to wait is a negative. Selvidge et al. (Selvidge) agree with this for the majority of cases, but their research led them to suggest that waiting can be a positive, especially for sites which are cognitively demanding. In these cases users use this time to process the information and to plan their next moves. This would support a slightly longer time delay in loading information in complex domains.

In general the wait time does not affect a user's opinion of the material, but it does increase the level of frustration. The level of frustration is likely to be affected by (Dellaert & Kahn 1999):

1. The length of the wait
2. The level of uncertainty of the wait
3. The level of information provided on the wait
4. When the wait occurs (e.g. at the start or in the middle)

Several pieces of research believe that the key to limiting this is to provide information on the delay or a progress report (Dellaert & Kahn 1999, Palmer 2002). Providing this information during the wait is able to reduce the level of frustration a user feels. However, the length of time should still not increase significantly. There does not appear to be any new research conducted despite the recent advances in broadband. King (King) notices that the average download time has decreased, but are users now only willing to wait less than 10 seconds for a web page?

2.6.3 CONCLUSIONS

The Internet is constantly evolving at a rapid rate, assumptions based on usability should be re-evaluated regularly as users perceptions adjust as the performance of the infrastructure increases. However, as infrastructure increases to enable quicker speeds so too does the complexity of applications thus requiring more information to be sent. Slow load times will frustrate the user leading to resistance to adoption. Research does suggest that waiting can be positive, especially for sites which are cognitively demanding. In addition it appears users will accept slight delays provided information/progress is provided during the delay.

2.7 MODEL DRIVEN ENGINEERING

The last decade has seen a growth in the discipline known as Model Driven Web Engineering. It aims at applying model driven engineering principles to web applications. This project aimed to utilise models with HTML, it was therefore considered suitable to review the field.

Model driven engineering advocates the use of models and model transformations as the key features in all phases of software development. It promotes the abstraction of models to be platform independent, with subsequent transformations to generate platform specific models for deployment. Many model driven engineering methods and tools also enable the automated generation of application code from these models. Model Driven Engineering has the potential to improve both productivity and maintainability (Hutchinson et al. 2011). There are many model driven web engineering methods, some prominent examples include, OO-H (Gómez & Cachero 2002), UML-based Web Engineering (UWE) (Koch 2000), WebML (Ceri et al. 2000), OOWS (Fons et al. 2003), WEI (Moreno & Vallecillo 2008).

All model driven web engineering approaches currently take a generative approach to producing the final web application. For dynamic applications, this includes the production of a data store either through generating a relational database based on the models or through transformations to the Ecore XMI and utilising the data through XML. Some approaches do make use of the Eclipse Modelling Framework (EMF) models through generating the Java model code and model editor classes. The BAE Systems development aimed to utilise the Eclipse Modelling Framework for the model development. Ecore is the meta-model used by EMF to produce models. The presentation coding is generated by all approaches, whether the output is PHP, JSP or static HTML. This level of generation results in several disadvantages. Any change to the meta-models requires the regeneration of the code. It also means that the data is duplicated in multiple forms. This leaves open the potential for synchronisation issues.

2.7.1 WHICH MODEL DRIVEN WEB ENGINEERING APPROACH?

Nearly all methods can be considered to consist of at least three platform independent models. These can be generalised to be, a concept model, a navigation model and a presentation model. Methods of transformations to platform specific models vary from the use of graph transformations like MIDAS (Cáceres et al. 2004) to template based, like WebML (Ceri et al. 2000). Predominantly the transformation languages QVT or ATL are used by the approaches for transformation between models (Koch 2000, Gómez & Cachero 2002). By validating the transformation, any subsequent output will produce valid standards-conforming HTML. It can be assumed that the use of models by some approaches for the design of screens reduces the chances of poor code standards by removing any programming necessity. The approach is also more usable to domain experts. It can also be assumed that only limited knowledge of web development is required to produce dynamic web applications, as rather than requiring code knowledge the developer simply builds the model. Knowledge of modelling and model management languages like constraints and transformations is all that is required. In addition constraints are generally written using the Object Constraint Language (OCL).

Several solutions make use of their own languages or tool applications. For example WebML utilises its own extensions of UML, and WEI defines a custom toolset (GlueWeb), which is, an incomplete subset of OCL combined with QVT (Moreno & Vallecillo 2008). This requires a large knowledge base of modelling languages to facilitate development as well as making interoperability highly difficult. Interoperability was identified by Vallecillo et al. (Vallecillo) as a significant flaw. To address this, solutions are being developed like WEI (Moreno & Vallecillo 2008), which has proposed a common reference meta-model to enable methods and tools to interoperate and can be used to define and build a web application, as well as integrate aspects from separate methods. However, the process is only partially automated with manual intervention required for combining models.

As the literature uses a model as either the data store or the template to generate a data store, it is believed that this will result in a storage solution that is geared explicitly to the information. This is in contrast to many web content management systems where the storage solution is geared towards the tool itself. A solution based on the information is likely to improve performance and storage efficiency. Those approaches which can utilise the model directly, such as UWE, further improve quality by reducing the amount of duplication as no database is required to be generated.

Development at BAE Systems has already resulted in a model of the data (See 4.5.1 & 4.5.2). This model is available in Eclipse Modelling Framework (EMF) according to the Ecore model standard (The Eclipse Foundation 2009c). Although model driven web engineering methods, in principle, offer the ability to improve quality, this must be traded off against the development effort. BAE Systems required a solution which could integrate with the work already developed, for a low level of development cost. As this was an early stage research within BAE Systems, cost drivers dictated that commercial variants were not considerable.

This results in several of the model driven web engineering methods being rejected for BAE Systems purposes. Koch et al. (Koch) provides a good review of case tools available as of 2008. They identified the lack of available tools, a position which does not seem to have changed. Some of the leading methods of model driven web engineering require the use of commercial products such as WebRatio (Acerbis et al. 2007) for webML and OlivaNova (Care Technologies Ltd 2010) for Object Oriented Web Solutions (OOWS). UML-based Web Engineering (UWE) appears reliant on the commercial product MagicDraw, however, an Eclipse plugin (Kroiss et al. 2009) has been recently made available. Eclipse is the software development platform that was under evaluation by BAE Systems for the model development work (see section 5.3).

As development cost needs to be low, the amount of additional implementation required can rule out many methods. For example, WEI requires the development of thirteen models, MIDAS (Cáceres et al. 2004) requires the development of five models and graph transformations. Similarly, WebSA (Meliá & Gómez 2006) would require the production of three models to handle the web architecture, this is in addition to the demands implied by using an approach such as OO-H or UWE to produce the functional viewpoint e.g. data model and navigation/process models. This would require the production of at least four new models with no supporting case tool. WebSA does have the benefit of being solely UML based and therefore usable within any UML compliant application. This is true for many methods such as UWE, OOWS and others. Out of the methods observed two stand out as potential lightweight solutions, OO-H and UWE.

OO-H provides a case tool VisualWade (VisualWade 2006). The method itself is relatively lightweight requiring only a data model, navigation diagrams and composition layout diagrams (the screen design). The method produces applications based on PHP and mySQL which are both proven technologies. However, it is apparent that the tool is out dated due to the use of dated pre-requisites and lack of recent information. An example of a navigation diagram is provided in Figure 2-6. In producing a demonstration using the tool, it became apparent it was not suitable. VisualWade requires the production of its own class diagrams which cannot be exported or imported. The class model also provides no method for handling inheritance. In producing the composition layout template it is also observed that there is only limited control over page design for lists of links. However, several tasks are automated which speeds development, such as the generation of basic pages given a navigation diagram.

UWE, is another relatively lightweight approach in terms of required models. UWE’s case tool however is based on MagicDraw which requires commercial licenses. A plugin for the Eclipse, UWE4JSF (Kroiss et al. 2009), has been recently released, but this appears to be geared towards the generation of the dynamic script and MagicDraw is still required for the implementation of the models. UWE however, is UML compliant and the meta-models are made available for use with any UML compliant software. UWE requires a process model in addition to OO-H. UWE’s main benefit over OO-H is the presentation class. The presentation class allows greater control over the design of an application than composition layout diagrams. The UWE navigation model is simply a stereotyped class diagram, unlike OO-H. Being UML compliant UWE is able to handle the multiple inheritances in the content model that OO-H is not.

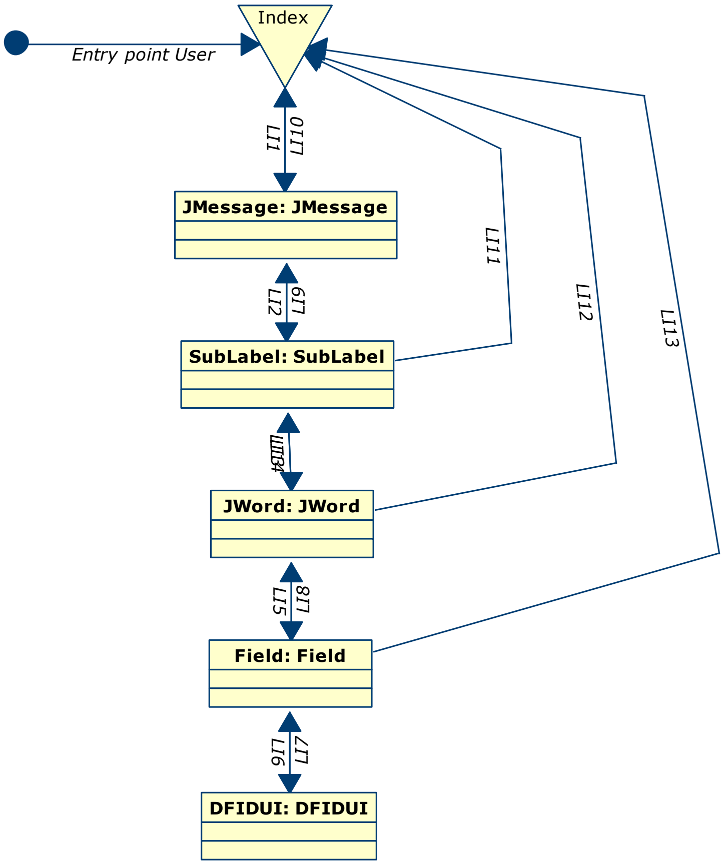


Figure 2-6: Example OO-H Navigation Diagram for TDL Message

2.7.2 CONCLUSIONS

Model driven web engineering approaches are either moving towards being solely UML based, which allows a user to implement the approach in any UML compliant software, or have developed a commercial product. The more recent approaches continue to increase in complexity and increase the required development effort. This is primarily driven by the focus towards model driven architectures and finding a solution which is platform independent. The domain will benefit from tools geared more directly towards their needs.

The principles they aim to achieve will provide methods to improve the quality of web applications. Either through using models (class models) for data structure or the basis of auto-generating database schemas, or by the ability to validate a transformation and thus rely on it to produce standards conforming code with no need to post-process for validity. The use of UML diagrams also makes developing dynamic web applications more accessible as model/domain experts can use their modelling knowledge to produce applications.

2.8 SUMMARY

This chapter has reviewed literature that has impacted on areas of this research. The literature review has found little research that fulfils the aims of this project. Although the Health Level 7 CDA has similar aims, its approach has the opposite problem. With Health Level 7 the sources are widely distributed and the aim is to standardise these to enable easily centralised access. The project described in this thesis already has a tightly controlled central source, the problem is more accessible distribution.

This project aimed to combine semantic models with document structure models. There are many document markup languages which define a documents structure, these have been reviewed. Although methods such as DocBook and DITA are designed primarily for technical documents they are considered excessive in overhead where a simplistic approach would provide the same ability for proof of concept exploitations whilst functioning within the business constraint R1 for rapid exploitation.

The literature recommends that an ORDBMS and/or ORM are used when assessing persistence. This project utilises both complex data and querying which is considered the area where ORDBMS are most effective.

As any exploitation was required to be through the use of pre-approved software (constraint R2), HTML was key to providing new functionality to engineers. Therefore, this chapter has reviewed literature to understand best practice for the display and navigation in electronic documents. In addition a review has been conducted to understand users’ expectations given the rapid development of Internet architecture.

Finally MDWE was reviewed, as the project aims to utilise models and web based rendering. An objective for this project was to reduce the number of sources of the material. MDWE at present utilises generative approaches to producing HTML, this would result in multiple sources to maintain. Although MDWE does simplify the process through automatic generation, someone must be responsible for ensuring this generation has occurred.

3 RESEARCH METHODOLOGY

CHAPTER PREFACE

This chapter discusses the various research philosophies, approaches and methods available to the researcher. It seeks to review and explain those common within information/computer science. It aims to provide a context for the research and identifies the process the author used to conduct this study in order to meet the aims identified in Section 1.3.

3.1 INTRODUCTION

The aim of a research methodology is to enable readers to establish the manner in which the researcher perceives interactions with the world and how they intend to direct their actions in conducting their research. In designing a research methodology various elements must be considered to establish this shared reference with the reader. Within social sciences, Blaikie (Blaikie) outlined a structure for considering the philosophical tenants for composing a research methodology, the main tenants being strategy and philosophy/paradigm. Blaikie highlights that the research question is informed by the strategy, whereas, the research methods are related to the choice of philosophy. It is then the researcher’s opinions on ontology and epistemology which drive the selection. This structure is provided in Figure 3-1.

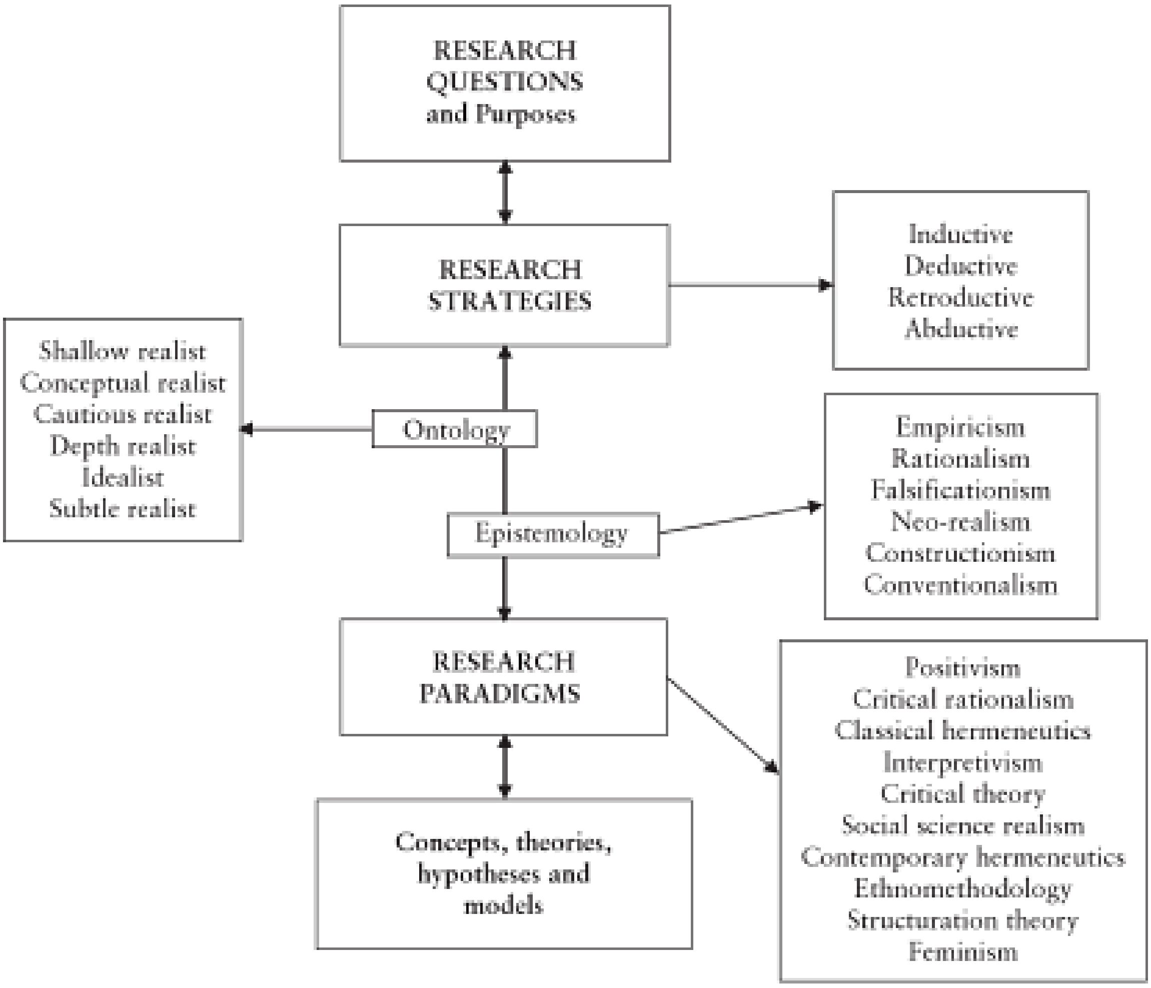


Figure 3-1: Research Strategies and Philosophies

Source: (Blaikie 2009)

Much literature does not differentiate between the elements ontology and epistemology, and philosophy (Guba 1990, Scotland 2012, Willis 2007). Guba indicates a philosophy can be characterised by its ontology, epistemology and methods. Saunders et al. (Saunders) do not differentiate with ontology, epistemology all being encompassed into a philosophy. They propose the concept of the research onion (shown in Figure 3-2) as a method structure for research methodologies. The layering of the onion provides the structure to building a sound methodology. Although their work originates from the discipline of business studies, it is just as relevant to computer science.

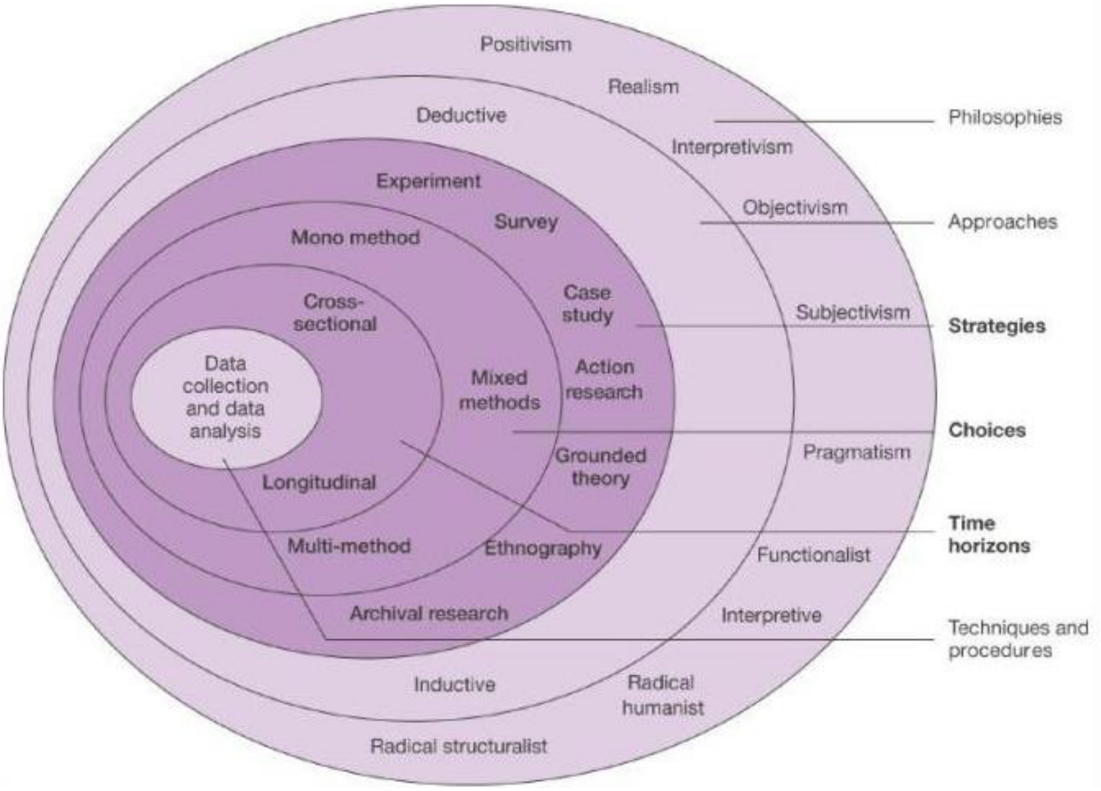


Figure 3-2: Research Onion

Source: (Saunders et al. 2009)

3.2 RESEARCH PHILOSOPHY

A research philosophy or paradigm defines an accepted shared belief or perception of the nature of the world and interactions with it. Bassey (Bassey) provides an elegant definition as "a network of coherent ideas about the nature of the world and the functions of the researchers which, adhered to by a group of researchers, conditions their thinking and underpins their research actions."

Within information system research there are three widely used philosophies, positivist, interpretivist (or post-positivist) and critical theory (Galliers 1991, Orlikowski & Baroudi 1991). Critical theory is, in itself, not a philosophy. Orlikowski and Baroudi (Orlikowski) define it as studies that challenge the established thinking through analysis, which attempts to reveal the historical, ideological, and contradictory nature of this thinking. Gabu and Lincoln (Guba) use it to encapsulate several alternative philosophies ranging from feminism to neomarxism. The three philosophies can also be extended further to consider constructivism (Guba & Lincoln 1994).

3.2.1 POSITIVISM

Positivism is based upon values of reason, truth and validity. The emphasis is on details gathered through direct observation and experience and measured empirically using quantitative methods. The position of positivism presumes the world exists objectively and externally. Any knowledge is only valid when based on observations of this external reality. It is generally characterised by the testing of hypothesis developed from existing theory through measurement of observable occurrences. Positivists believe that phenomena should be isolated and that observations should be repeatable. Often reality is manipulated with variations in only a single independent variable. This manipulation along with the repeatability enables researchers to identify regularities and to form relationships. Predictions can then be made based on the observed and identified relationships. Hirschhiem (Hirschheim) identified 5 characteristics of positivism.

1. Unity of the scientific method.
2. Search for human causal relationships.
3. Belief in empiricism.
4. Science (and its process) is value-free.
5. The foundation of science is based on logic and mathematics.

Methods associated with this positivism include experiments and surveys where quantitative data is the norm. Analysis methods using statistical or mathematical procedures are frequently used.

3.2.2 INTERPRETIVISM/POST-POSITIVIST

Interpretivism is highly contextual, hence is the opposite of positivism within the concepts of generalisation and the repeatability (Robson 2002). The focus is more on understanding the meanings and interpretations of actors/objects from their world viewpoint. As such interpretivism is subjective in nature and is seen as inductive or theory building. It is normally associated with the qualitative data collection. The close nature of the researcher and the phenomena introduces the risk that bias can be introduced due to interpretation framed within the mind of the researcher.

Interpretivists believe that reality can only be understood through the subjective interpretation of and intervention in that reality. Interpretivists acknowledge that scientists cannot avoid affecting the phenomena. They contend that there may be many valid interpretations of reality and that each interpretation is part of the overall scientific knowledge of the phenomena.

3.3 RESEARCH APPROACH

In identifying a research approach researchers typically focus on two areas, deductive and inductive, or idiographic and nomothetic. Idiographic and nomothetic are focused on collectively identifying methods of research whereas deductive and inductive tends to identify the logical approach to drawing conclusions.

A deductive research approach is often associated with the positivism philosophy (Crowther & Lancaster 2009). It allows the researcher to establish a hypothesis by using theory. The process generally followed is shown in Figure 3-2. A deductive approach aims to confirm or reject the generalised theory through manipulation of data and reassertion. An inductive research approach is more closely associated to the interpretivism philosophy (Crowther & Lancaster 2009). It has no requirement of pre-determined theory to collect data and information. The process generally followed is shown in Figure 3-3. The researcher observes the phenomena in detail and aims to define a theory using observations.

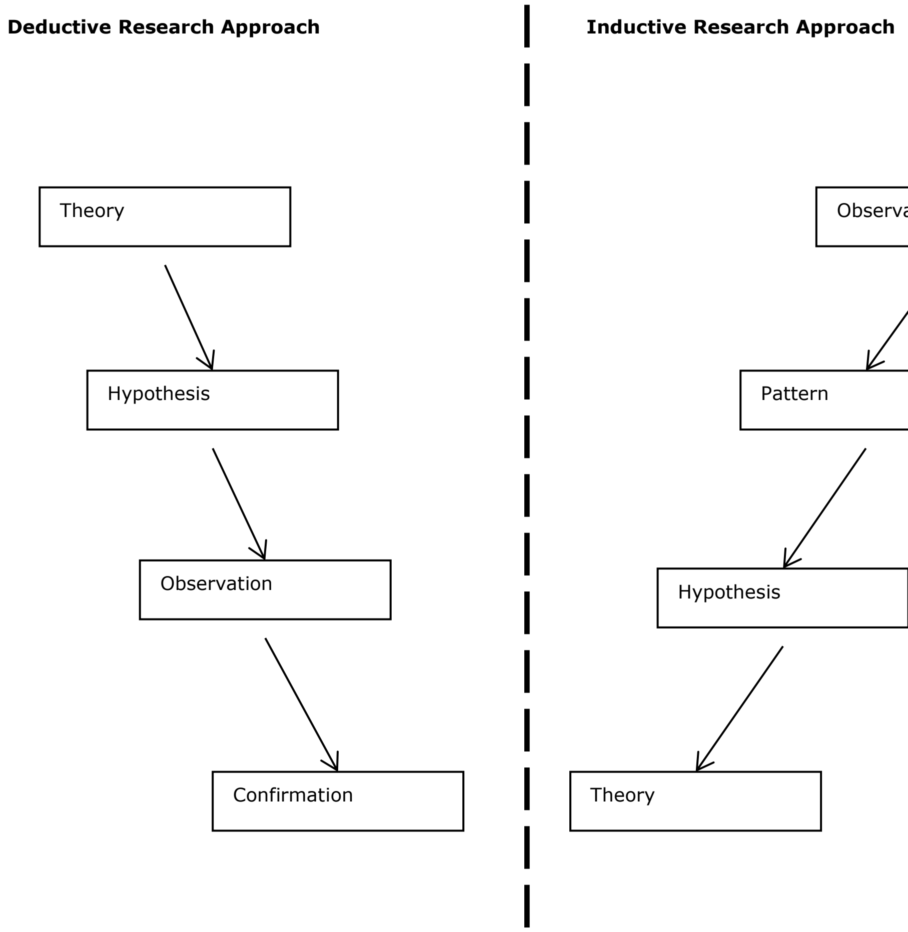


Figure 3-3: Comparison of Deductive and Inductive Approaches

Burrell and Morgan (Burrell) distinguish two types of research methods: nomothetic and idiographic. Iivari (Iivari) extends this within the concept of computer science to encompass a third category, constructive methods. Constructive methods are concerned with the engineering of artefacts, either purely conceptual (models, frameworks, and procedures) or more physical, e.g. software. Nomothetic methods are mostly aimed at generalisation and are generally deductive by nature. Nomothetic methods are usually highly structured and can be controlled and replicated focusing on quantitative data. Examples of nomothetic methods include laboratory experiments and mathematical analysis. Idiographic methods are mostly aimed at specification and are generally inductive by nature. Idiographic methods are typically qualitative. Examples of idiographic methods include case studies and action research.

3.4 RESEARCH STRATEGY

There have been several studies which document the various research methods/strategies (Galliers 1992, Glass et al. 2004, Holz et al. 2006, Mingers 2003). Of these, the most comprehensive study was performed as part of the ACM Special Interest Group on Computer Science Education (SIGCSE) identifying 54 methods used in computer science (Holz et al. 2006). The identified methods either built upon the taxonomy by (Glass et al. 2004) or were identified by the working group and supported by literary references. Not as comprehensive is Choudrie and Dwivedi (Choudrie), who relate the methods of Galliers (Galliers) and Mingers (Mingers) to a respective research philosophy of positivist or interpretivist. It is beyond the scope of this review to provide details of all 54 identified methods, readers are invited to read Holz et al. (Holz) for full details. A selection of methods that are used in this research project are provided in Table 3-1.

Table 3-1: Research Methods Descriptive Quotes

Research Method	**Descriptive Quote**	**Source**
Action Research	"seeks to bring together action and reflection, theory and practice, in participation with others, in the pursuit of practical solutions to issues of pressing concern"	(Reason & Bradbury 2001)
Case Study	"Single Case: examines a single organization, group, or system in detail; involves no variable manipulation, experimental design or controls; is exploratory in nature." "Multiple Case Studies: as for single case studies, but carried out in a small number of organizations or context."	(Alavi & Carlson 1992) See also (Benbasat et al. 1987)

Research Method	**Descriptive Quote**	**Source**
Document Analysis	"includes examination of system software and documentation, project technical papers and memoranda…"	(Salminen et al. 1997)
Evaluative Research	"involving methodologies that employ the scientific method, and usually consisting of theory or model generation or observation followed by hypothesis generation and testing."	(Morrison & George. 1995)
Grounded Theory	"[aims] to develop a theory from data rather than gather data in order to test a theory or hypothesis. This means that qualitative methods are used to obtain data about a phenomenon and that a theory emerges from the data."	(Goede & de Villiers 2003)
Interview	"an information gathering technique whereby people are posed questions by an interviewer; these interviews may be structured or unstructured."	(Benyon et al. 2005)
Mixed Methods	"the use of both quantitative and qualitative methods within a single study."	(Lister 2005)
Proof of Concept	"a claim about the value of a system design (or the design of a part of a system) is validated by building a system based on that design. Typically, the system that is built is not fully featured, but has enough functionality to convince the readers that the design can be effective. The proof-of-concept system is usually measured for performance or usability, to show that the new design is not so bad as to be unworkable."	(Holz et al. 2006)

3.5 ADOPTED RESEARCH METHODOLOGY

Within Organisation Science and Information Systems research, interpretive research was most commonly used until the late 1970s (Vreede 1995). In 1992, a review of 902 IS research articles found that all the empirical studies were positivist in approach (Alavi & Carlson 1992). It has often been observed that no single research methodology is better than any other methodology (Benbasat et al. 1987). Although most researchers do either quantitative or qualitative research work, some have suggested triangulation methods by combining one or more research methods in the one study to improve accuracy (Kaplan & Duchon 1988, Mingers 2001, Mingers 2003).

It was intended to produce a framework that would enable the concept of hybrid semantic-document models to be applied to multiple sources. The framework in this context refers to a collection of model components with guidance on where and how they can be used to construct a hybrid semantic-document model of a whole document. To achieve this, action research was utilised with the main method being a case study based on Tactical Data Links standards. Grounded theory (Goede & de Villiers 2003) is the discovery of a theory through the systematic analysis of data. This research started with the concept of hybrid semantic-document models and, by using a grounded theory-like approach the aim was to discover the relationships and means of representing these relationships from the MIL-STD-6016C case study to produce this framework.

These methods were applied and evaluated using an interpretivist philosophy. An interpretivist philosophy was employed given the multitude of potential valid interpretations of the data and formulation of frameworks. Within this, a constructive approach as per Iivari (Iivari) was used to define the framework. This utilised an inductive process by observing the material present in the Tactical Data Link Standards and using these observation to infer a framework which aimed to achieve the objectives in Section 1.3.

In conducting the case study, a significant form of data capture was required. To obtain this data a document analysis method similar to that of Salminen et al (Salminen) was used. They use multiple sources as part of their document analysis, however, given the volume of MIL-STD-6016C and its segregation into discrete sections it was considered sufficient for data gathering. Given the volume, sampling was required. The sampling method was performed using cluster sampling. Given prior knowledge of the document’s format and structure, random samples were picked to highlight the types of document and information likely to be encountered.

The case study method has been used to obtain evidence to justify the decision taken in constructing the framework. By using the case study method, chapter 4 is able to review existing sources of material for the domain to obtain a benchmark. Access to actual TDL engineers was not possible, hence an understanding of the existing material allowed for the evaluation of the new approaches. Chapter 5 has conducted a series of performance tests using the case study domain to assess usability against existing source reviewed in chapter 4 and literature. Chapter 6, utilised the case study with document analysis to construct the proposed model, this has then been evaluated against the aims in section 1.3. Finally, chapter 7 has proposed an enhancement to the framework constructed in chapter 6 by applying the framework to the case study material and reviewing the effects.

3.6 SUMMARY

This chapter presented a detailed overview of the formulation of a research methodology including philosophies, approaches and strategies for research. Both positivism and interpretivism are common philosophies within the information/computer science realm. A description is provided for inductive and deductive as well as the nomothetic and idiographic approaches. A brief description of research methods/strategies identified by the ACM special interest group in computer science education were provided, before finally outlining the author’s methodology of a case study using an interpretivist and constructive approach.

4 IMPROVING USABILITY

4.1 CHAPTER PREFACE

This chapter reviews the document chosen to be the case study for this research (MIL-STD-6016C) and its representational formats such as Word and PDF. In particular it will look at features such as the usability and navigability of these representations of this standard. It details improvements made to navigation through the extension of a BAE Systems prototype application to achieve objective A1 (Improve navigation). It includes material published in Paper 3 (See Appendix C), Paper 4 (See Appendix D) and additional paper 2009 (Clowes et al. 2009).

4.2 INTRODUCTION

A primary objective for this research was to improve the usability of complex technical standards. To improve the usability it is important to understand what is already available, its important features and its limitations. A key area of usability when using large documents such as the MIL-STD-6016C is the ability to locate, recall and navigate the information.

4.3 EXISTING SOURCES AND REPRESENTATIONS

The MIL-STD-6016C is published by the United States Department of Defense (DoD) in both Adobe PDF and Microsoft Word format. The MIL-STD-6016C is currently only available to engineers in electronic format. This is mainly due to the 7300 pages being equivalent to approximately 15 reams of paper, or enough to fill two entire filing cabinet drawers. The review of literature in Section 2.5.1 argues that it is impractical to use paper based versions of large documents, given their weight and size. This is certainly true for the MIL-STD-6016C, however, engineers may choose to print out particular sections that they are working on.

In electronic format the DoD provides the standard in an Adobe PDF file or a collection of Microsoft Word files. It is important to note that the standard is not available in a single Microsoft Word document. The standard is spilt over 155 separate Word files which, when combined, represent the complete standard. As the standard defines all requirements for all roles of platforms, projects develop their own project version of the standard, which in most cases is just the platform required subset of the base standard. As such, this project specific version is often maintained using only Microsoft Word. This research used the base standard MIL-STD-6016C as this does not restrict the work to one project. Before describing the features of the pilot application, the advantages and disadvantages of the MIL-STD-6016C in its current available formats are analysed.

4.3.1 MICROSOFT WORD

Within Microsoft Word the navigation is primarily by scrolling. A hierarchical menu can be generated through the document map function. However, the document map functionality requires the consistent use of Microsoft Word Styles such as heading 1 to generate the menu. The MIL-STD-6016C has no consistency in its textual formatting, in places it uses fixed width characters and spaces, where in others it uses formatting aids such as tables. Because the standard is spilt over several files the generated menu only works for that specific document.

Within the content area there are a number of non-linear links which jump to the relevant section in the relevant document, opening the file if needed. In creating the project specific document, the projects only use one file, however, they simply copy over the required sections of text from the MIL-STD-6016C. Microsoft Word also allows for page jumping through the “Go To” option available via the edit menu/find option of the editing group in the home tab depending on version. When scrolling, the page number is displayed during the process to indicate the position. The scroll to find a page becomes increasingly more difficult as the page count increases in documents over 100 pages. Word also supports linear linking with its next and previous buttons. An illustration is provided in Figure 4-1 to show the main navigation aids and their approximate spatial location.

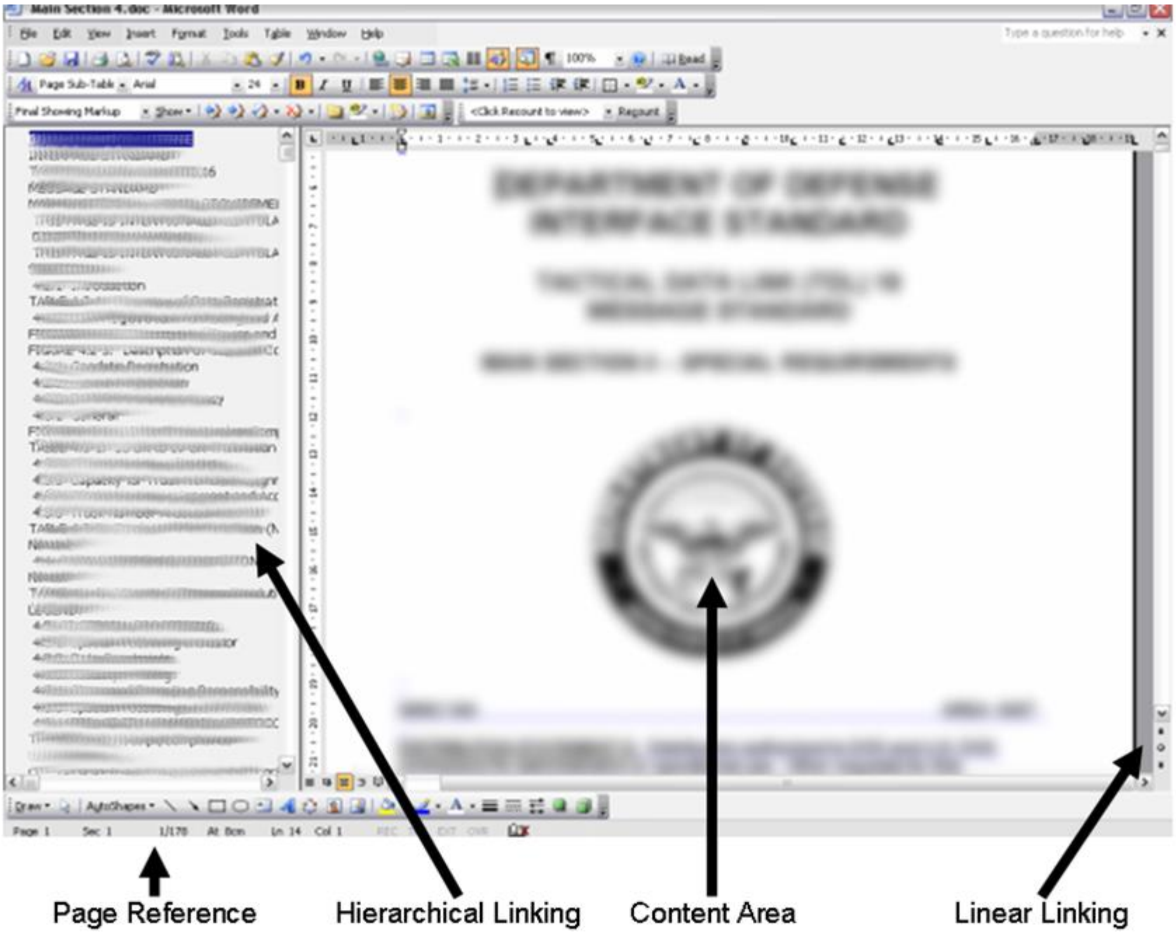

Main Section 4.doc - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Page Reference
Hierarchical Linking
Content Area
Linear Linking

Figure 4-1: Microsoft Word Navigation aids

(Picture distorted for security)

4.3.2 ADOBE PDF

Adobe PDF is the other format that the TDL Link 16 standard is widely distributed in. Unlike Word, the whole standard is available in one file. Within this format it makes use of bookmarks to provide a hierarchical menu. The hierarchical menu has been constructed with three top level categories, “Main Sections”, “Appendices” and “Concluding Material”. The document creators have then limited the sub-level to a further two headings. This means the lowest level that can be navigated through this hierarchical menu is the section plus the main subsections (e.g. 4.2). The standard itself uses headings down to 4 levels (e.g. 4.2.1.1).

The Adobe PDF format makes use of both paging and scrolling with either technique being just as easy to use. Page jumping is easy, simply entering the required page number in the page reference information which is more emphasised than in Microsoft Word which, according to section 2.5.2 and, in particular, Piolat et al. (Piolat et al. 1997), should help limit disorientation.

The Adobe PDF format has limited non-linear linking. Again, due to how the document was created, in the contents list, the top level sections are linked to the position in the document, however, within the document the text is styled as a link, but no functionality is behind the text, which is quite confusing for the user. This is a limitation in how the document has been produced and not Adobe PDF, however, BAE Systems do not have the ability to regenerate the PDF to include these links. A diagram is provided in Figure 4-2 to show the main navigation aids and their approximate spatial location.

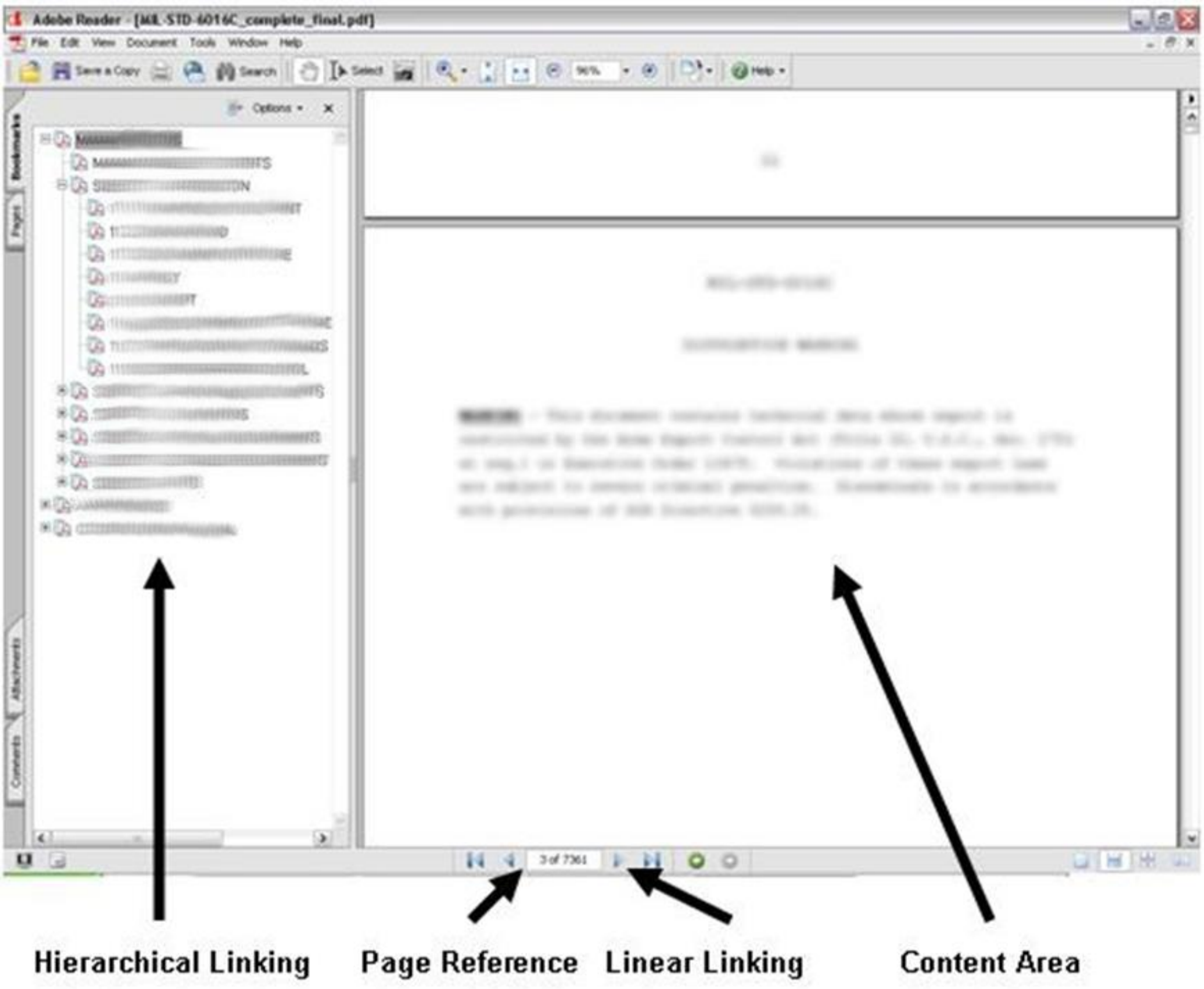


Figure 4-2: Adobe PDF Navigation aids

(Picture distorted for security)

4.3.3 EVALUATION

The Microsoft Word version of the MIL-STD-6016C standard is possibly the worst format to use whilst reading or searching for information. Microsoft Word does have the functionality to produce a hierarchical menu through the document map function, but this is limited to the document itself and, again, having the standard split over several documents means the menu only displays the information in the current file. In addition, the document map function appears to rely on consistent formatting using the in-built styles in Microsoft Word, which in the standard is not always guaranteed. On the positive side, Microsoft Word does have non-linear linking enabled within the document whereby the corresponding file to the link is opened on clicking, which the Adobe PDF file does not.

The Adobe PDF format has the significant advantage over Microsoft Word in that this version of the standard is in a single file. This enables the Adobe PDF format to have a hierarchical menu covering the whole standard. However, this menu is limited only to three levels. The Adobe PDF is the best format for moving to specific, already known pages (via the page number).

4.4 BUSINESS REQUIREMENTS

Two critical requirements/constraints (R1 & R2) imposed on this project by the business are documented in Section 1.3.3 Business Requirements & Constraints. The team investigating model driven technologies was motivated to get end user support by making any tools rapidly accessible to the users for both feedback and research buy-in to meet requirement R1 (Rapid prototyping).

Being in the defence industry, Information Technology is highly regulated internally, this led to the constraint imposed in R2 (Approved software). Any new bespoke application cannot readily be deployed to end users even in a sandboxed environment. Additionally, any new application that would need copying or installing on machines must first have passed through strict time-consuming and expensive internal security and validation checking. This makes small, rapidly evolving research platforms very difficult to exploit and generate sufficient buy-in to make adequate justification to instigate the development of applications.

For these reasons, any discovery/method needed to be exploited by using software already available to engineers. This, in reality, meant a standard Internet browser. By using a standard browser, there would be no need for software installations by the target engineers resulting in easier and quicker rollout to engineers. It would be relatively easier to maintain a development server through which, within certain classification areas, users would be able to access and utilise the standard.

4.5 EARLY WORK

Within BAE Systems, projects were already looking to address some of the shortcomings of the Word and PDF representations of the standard, by considering alternative ways in which the standard could be represented. There was no central research team investigating a shared approach, each project was attempting to improve the utilisation of the material through their own efforts. This led to several different approaches. The techniques and tools utilised ranged wildly from Matlab, Microsoft Excel, and Mandril (Lockheed Martin 2012) for navigating and parsing JMessages to IBM Rational DOORS (IBM Corporation 2012) to enable some form of traceability between the standard and the regulatory required platform documents based on the standard. This was to aid users maintaining quality by using the traceability to ease quality control of missing elements and comparison. Discussion with project team members established that just over half the project teams involved in this study were using DOORS with the majority of the other project teams envisioning using this software in the future for TDL documentation. Given the business constraint R2 (Approved software), DOORS would be available to the majority of users as well as an Internet Browser.

A central research team was formed within the System Engineering Innovation Centre (SEIC) independent of the projects. The team, at first, was unaware of the work being undertaken by projects. Research by the SEIC started in 2005. The research project of this thesis, hybrid semantic-document models, was started in 2007 with the objective of joining this SEIC team to aid in the development and extension of their work. The BAE Systems SEIC team had focused on the development of semantic models of the information and a document model of the structure.

4.5.1 SEMANTIC MODELS

An analysis of the Link 16 TDL standard by BAE Systems (Holmes & Johnson 2005, Holmes et al. 2007) led to the belief that the domain may be captured by a hierarchically layered set of meta-models, the lower two layers of which comprise the Data Dictionary and Message Catalogue as illustrated in Figure 4-3. This hierarchy of models is referred to colloquially as the *semantic models*. An excerpt of the Message Catalogue can be observed in Figure 4-4. Modelling commenced in mid-2005 using Xactium's XMF Mosaic tool; this is a powerful graphical meta-modelling tool underpinned with a fully-fledged programming language based on the Object Constraint Language (OCL) (Object Management Group 2006) called XOCL, allowing the tool to support the development and transformation of executable (meta-)models. It was an early requirement to be able to re-render material from the *semantic models* in a format similar to that of the source material (in this case the standard). Discussion of the semantic models developed for MIL-STD-6016C is outside of the scope of this thesis and is subject to security classification and pending patents.

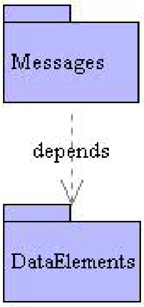


Figure 4-3: Data Dictionary and Message Catalogue Relationship

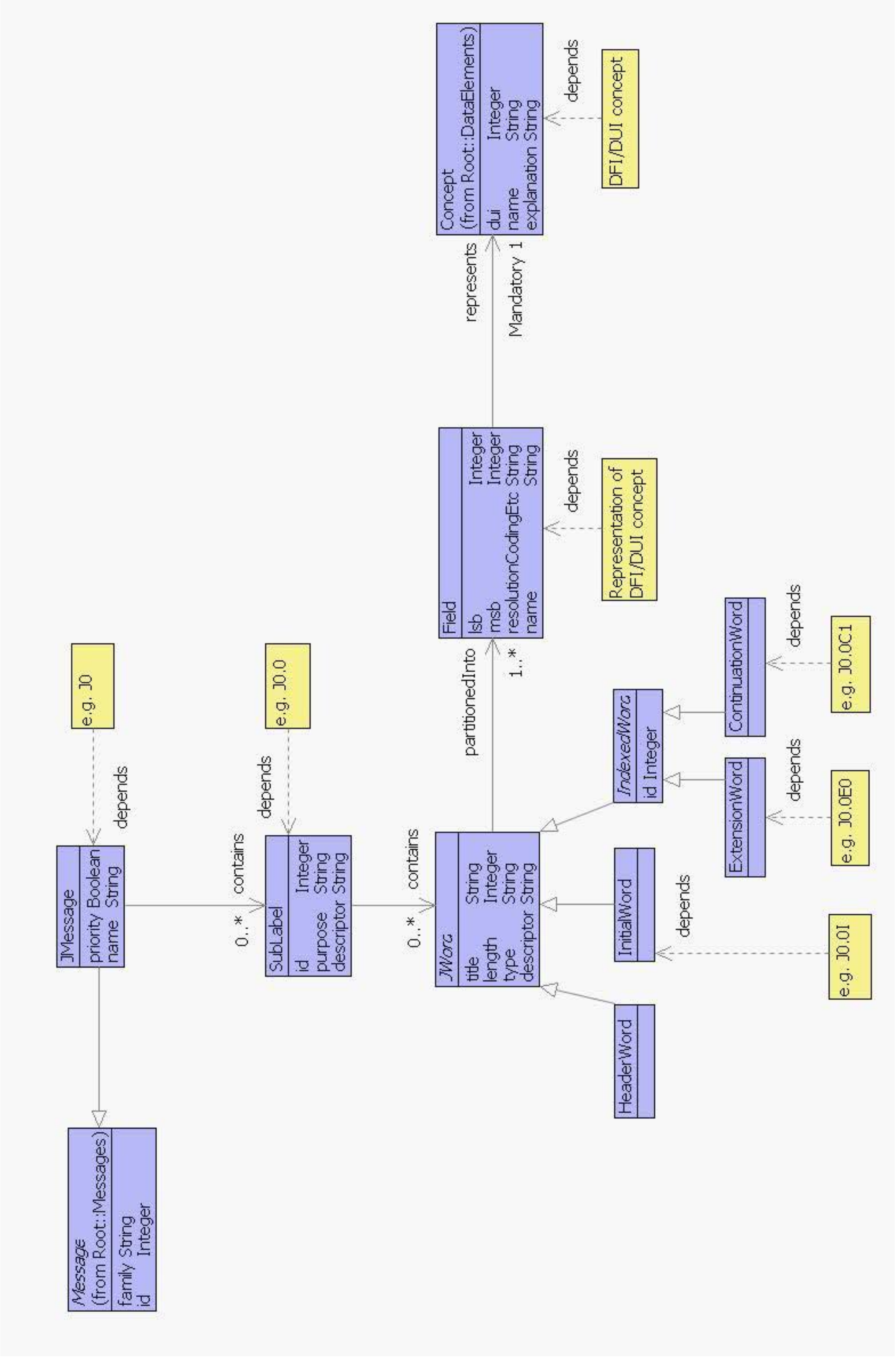


Figure 4-4: Message Catalogue Excerpt

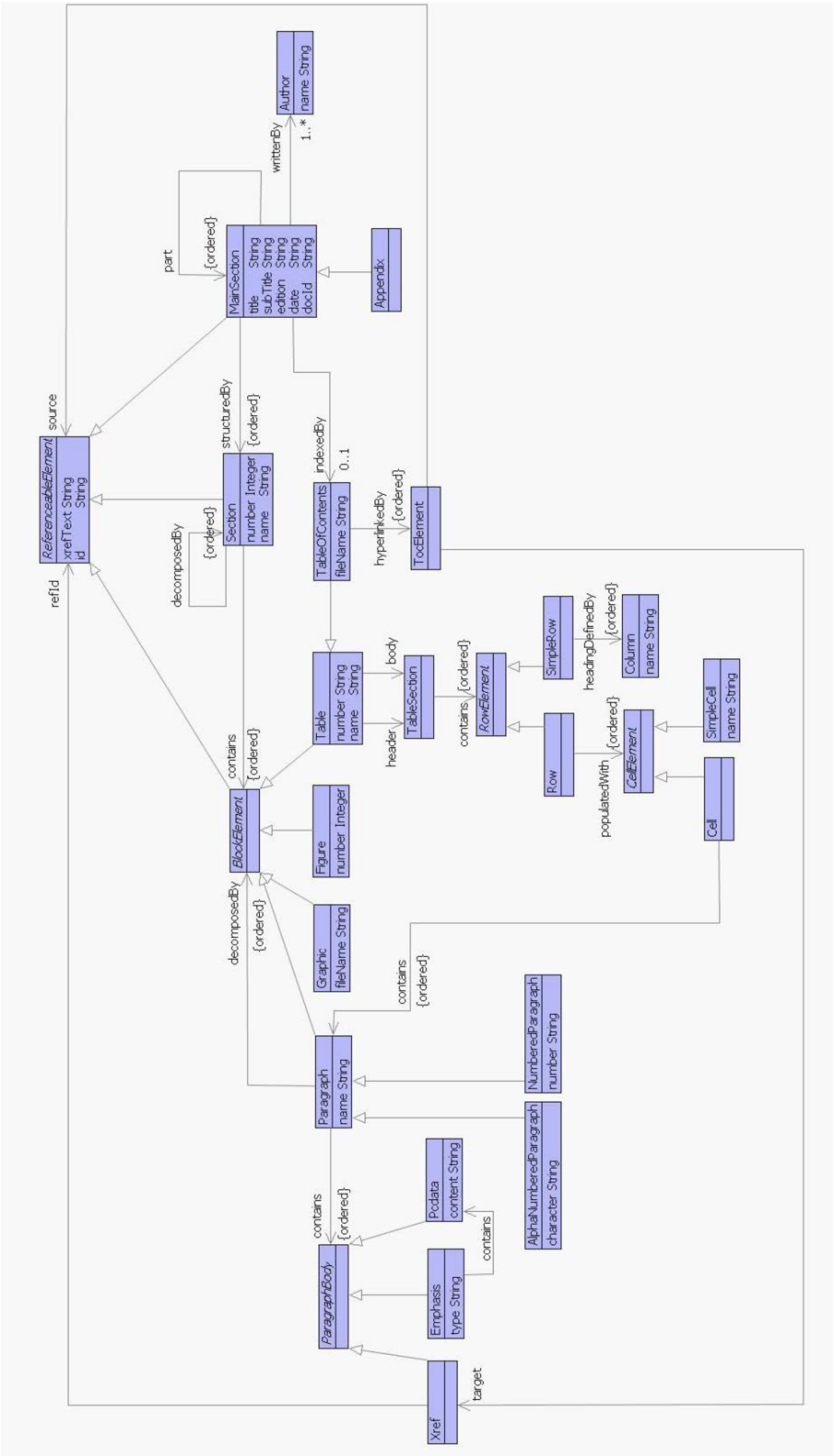


Figure 4-5: BAE Systems document model

4.5.2 Document Model

A document model was also developed, which allowed the standard to be captured and stored in a relational database. The aim of the document model is to be able to represent the standard in a format familiar to the engineers through HTML (e.g. standard prose and tabular). Figure 4-5 shows the document model used. As can be seen from the figure, the model at the highest level consists of a set of main sections, which in turn consists of sections made up of block elements. The model allows for any object at these three levels to be referenced. Worth noting is that a table cell element is made of paragraphs which can be referenced, leaving the possibility to reference individual cells. In addition the model has no concept of a page.

4.5.3 Interface

The SEIC research team had selected Microsoft SQL Server and Microsoft ASP.NET as their technology choice for the development of an interface for utilising these models. The team had written document parsers to generate an XML version of subsections of the MIL-STD-6016C. This was then imported into SQL Server to enable them to render the document according to their document model. The team had successfully managed to render the document by using a 15 element concept of a page. The semantic models were not available within the SQL Server.

4.5.4 Evaluation

Early research into both semantic and document models at BAE Systems was disjointed. The SEIC aimed to centralise and concentrate research amongst the projects teams. Their research was at an early stage of development at the time this Engineering Doctorate project was incorporated into their team.

The interface the team developed for rendering of MIL-STD-6016C was not designed to facilitate rapid prototyping. The design and development utilised ASP.NET with no structured architecture. Applying an architecture, such as Multi-tiered, enabled rapid delivery and exploitations aiding conformance with Business Constraint R1 (Rapid prototyping). In addition, this separation of presentation, logic and storage would potentially make facilitating interfaces with other applications such as IBM Rational DOORS easier.

The interface started by BAE Systems (described in Section 4.5.3) demonstrated that representing the standard over HTML was possible. In addition, it improved the navigability of the standard compared to MS Word and Adobe PDF. Similar to Adobe PDF, the material is now located in a single source and not spread across multiple documents. The level of navigable menu links was extended beyond the three levels supported by Adobe PDF enabling easier depth navigation. In addition, text styled as hyperlinks in the standard is now an actual hyperlink allowing the user to simply click the link rather than search for the reference.

However, the document model utilised, did not contain a concept of the page. This meant that the rendering of information between the current sources (Word and PDF) and this HTML view were different, this makes recalling the information location more difficult. Given the volume of the material, it is considered unlikely that the engineers recall the source of the information based on page reference only. This, coupled with the ability to extend navigation, lead to the belief that this should not cause an issue for the engineers. The document model is nevertheless very useful in proof of concept work given the simplicity of the model. Alternative models could be used such as DocBook (OASIS 2009), DITA (OASIS 2010) etc., however, these could easily over complicate the research and therefore conflict with business constraint R1 (Rapid prototyping).

The use of SQL Server was presenting the team challenges, the data needed to be stored and manipulated in many different formats. Their semantic models were designed using Mosaic XMF, which did not read from SQL Server, hence to use the semantic model for any model driven engineering task, the data needed to be maintained in multiple sources leading to consistency issues. In addition the semantic models are highly object oriented, whereas, SQL Server is relational oriented, some of the complex object designs such as multiple inheritance are difficult to replicate with a relational database. Therefore, the early work did not make any utilisation of the semantic and document models combined.

4.6 INTERFACE PROTOTYPE

One of the assumptions of the Engineering Doctorate for the SEIC research team was to enhance the work already conducted. The document rendering application was thus extended. A key business requirement R1, is the ability for rapid prototyping. The application described in Section 4.5.3 had been developed with no architectural underpinning, so it proved difficult to enhance with rapid changes. An early enhancement was to re-engineer the application into an N-tier architecture (Clowes et al. 2009). This architecture, by being multi-tiered, aimed not only at supporting rapid prototyping, but also potentially enabling integrations with IBM Rational DOORS, hence allowing for the support to multiple tools.

Using the models, a new interface was developed for viewing the standard which can be seen in Figure 4-6. This representation became known as the document navigator. The application makes use of many of the navigation features advised in the literature review in Section 2.5.2. A breadcrumb trail is provided to aid a user's awareness of their location within the document. The new application does not allow for scrolling of the document, either the hierarchical menu and/or the linear links (previous and next page buttons) must be used. This is to limit the amount of data transmitted thus enabling acceptable page load times. Elements enhanced by the earlier BAE Systems work (Section 4.5.3) have been maintained, as the model does not contain a concept for a page, a value of 15 block elements has been maintained and set as a page. This results in a flexible page length, as not all block elements use the same amount of space, unlike the PDF and Word formats. The hierarchical menu is not constrained to a fixed depth, allowing users to drill down to the block element level provided they have a corresponding referenceable element within the model. Within the content area, non-linear linking around the document is functional.

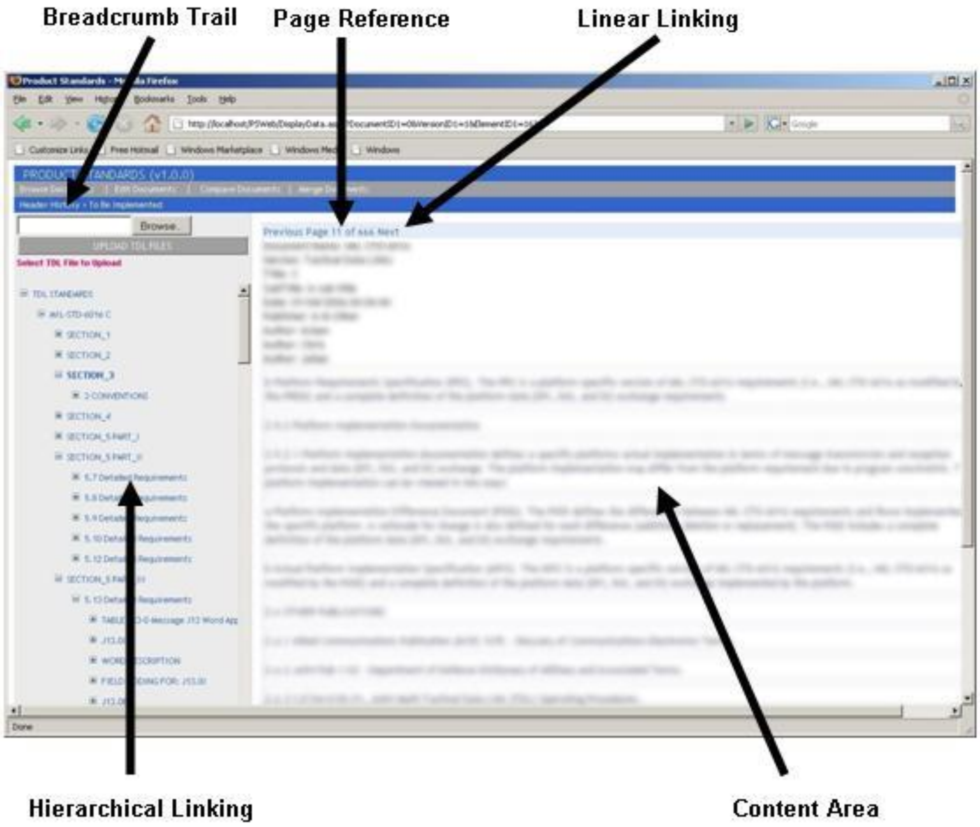


Figure 4-6: Experimental HTML Document view

(Picture distorted for security)

The document navigator was developed as a simple rendering of information within the document model. This interface was first developed using ASP.NET and Microsoft SQL as the data source. The technology is not considered relevant in this chapter as the interest is in the navigation.

Additional interfaces were developed to support the information captured in the Message Catalogue (Figure 4-7) and Data Dictionary (Figure 4-8). These interfaces used the semantic model to enable new navigational routes through the material. Users could navigate up/down through the hierarchy and follow links between the Data Dictionary and the Message Catalogue, this navigation provided an alternative to moving between the appendix tables to find the information.

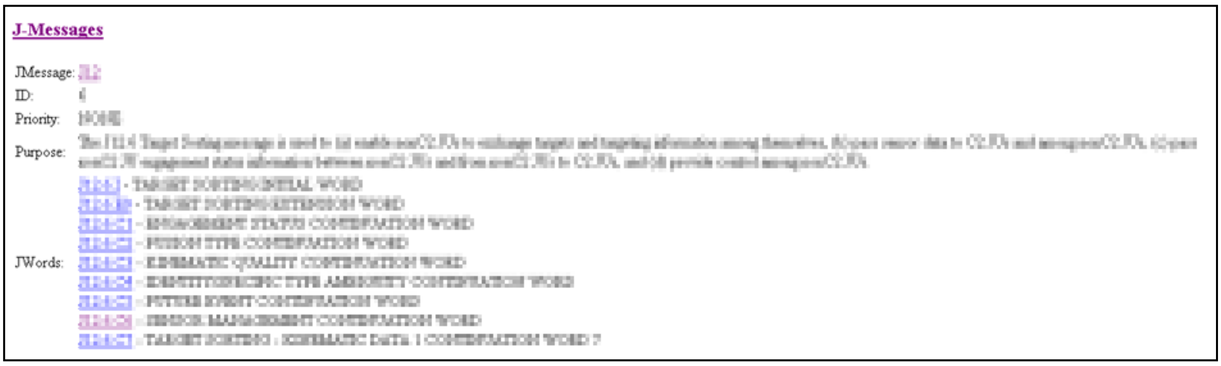


Figure 4-7: Message Catalogue Viewer

(Picture distorted for security)

Figure 4-8: Data Dictionary Viewer

(Picture distorted for security)

4.6.1 Evaluation

The HTML prototype described in Section 4.6 has the potential to use the benefits of both Microsoft Word and Adobe PDF whilst trying to limit their disadvantages. It does allow for a full depth hierarchical menu to be generated and used which helps achieve objective A1 (Improve navigation). The extended version implements many of the navigational aids identified in literature (Section 2.5). As well as providing new aids, such as a breadcrumb trail, the new Message Catalogue and Data Dictionary interfaces provide alternative methods to rendering and interacting with the appendix. This area in the existing sources is an implicit link, in that, the text reads the identifier of the corresponding object; the user must manually navigate around the standard to gain the detail. The new interfaces automate this and allow for a complete navigation both up and down the hierarchical structures of the Message Catalogue and Data Dictionary significantly providing functionality to meet objective A1 (Improve navigation). In addition, the two models are interlinked allowing for quick navigation between the two structures where a relationship exits. To evaluate what performance benefit this is to users, a sample of 89 messages from the standard were selected and using both the PDF and Microsoft Word versions of the document, they were navigated with two tests.

Test 1 aimed to determine the amount of navigation required simply to obtain a list of JWords for the given JMessage. The PDF file was easy to use, in that the hierarchical menu allows the user to navigate directly to the JMessage description with five steps. Only 18% of the JMessages required the user to navigate onto another page for the continuation of the JWord data. The Microsoft Word file was not easy to use. The user had to navigate ten pages to get to the index list entry for the correct section of the document. The index entry then loads another JWord file, from which an index link is followed to another part of the document which tells the user to find three further files which are not directly linked to the document. There are, therefore, three index files which link to further files containing the information.

Test 2 aimed to discover the amount of further navigation required to determine the deeper detail in the JWords data element descriptions. This test required the same amount of navigation as the previous test plus an average of 33 linear page moves to generate a list of Data Element descriptions for a given JWord. The results are shown in Table 4-1.

Table 4-1: Number of pages to navigate based on PDF

Page Moves	Number of Words
0-10	29
10-30	37
30-50	9
50-100	5
100+	9

The new navigation interfaces reduced the number clicks irrespective of the message. By allowing the HTML systems hierarchical menu to traverse down lower than three levels, the document model version also allows for direct navigation of the JMessage structure due to the presence of defined referenceable subheadings in the hierarchical menu. These new interfaces provide proof of concept evidence to the amount of navigation increase that can be achieved. This improved navigation is also expected to aid users understanding.

The behavioural semantics, however, are significantly more complex than a hierarchical structure. One of the simplest cases is that of pairing of tracks in the Weapons Co-ordination section. The pairing message is defined in J.10.6 which can be found on page 1838. To establish the behaviour of pairing messages the engineer must then analyse the transmit and receive tables. These are found starting at page 3422. The engineer must then move to page 3433 to establish that the Pairing is defined in Appendix K and has two transactions K.6.1 and K.6.2. The appendixes are located on page 5852. The appendixes define the constraints and stimulus for the transaction to occur. The actual values are located in the transmit and receive tables. The transmit data is defined on page 3642, and the receive on page 4154. Other instances, such as J12.0 or Mission Assignment, are a lot more complex requiring even more movement and cross-referencing.

Although no interface was produced to navigate the behavioural model aspects, interfaces similar to the message catalogue and data dictionary could easily be developed to traverse this structure, greatly improving the location of information in this area. Such an interface removes the need for a lot of the cross referencing and massive page jumping and the model can provide links to the relevant section of the document, for example, providing appendix K.6 with the J10.6 transmit and receive table as a section.

4.7 ASSESSING PROTOTYPE PERFORMANCE

The prototype HTML presented many navigation enhancements, however to achieve an enhancement in usability, the application must also perform to an acceptable standard whilst providing this navigation improvement. This section analyses the performance of the prototype.

4.7.1 EQUIPMENT

The experiments were carried out in a standalone environment to rule out potential performance issues with the creation of a network. They were run on a Toshiba Satellite Pro laptop running an Intel Core 2 Duo T7300 CPU running at 2.00GHz. The machine had 3GB of RAM and the Hard Disk Drive was a Toshiba MK1637GSX 2.5-inch 160GB running at 5400rpm. The operating system was Windows XP Service Pack 3 which was also running Microsoft Internet Information Services (IIS) 5.1. The SQL Server was Microsoft SQL Server 2005 standard edition using shared memory. All unnecessary services and applications were shut down on the machine during the tests including virus scanners etc. The web browsers installed included Microsoft Internet Explorer 7 and Mozilla Firefox 3.

4.7.2 BENCHMARK AND TEST PLANS

The Adobe PDF version of the MIL-STD-6016C standard takes between two and five seconds to load. This ignores any amount of time to download the file. The file itself is 15Mb. A perfect solution would have load times comparable to this. However, as literature in Section 2.6.2 suggests, users are prepared to wait longer for web page loading, as such the target set for an acceptable interaction was 10 seconds. It was also decided that the web-based solution would be classed as acceptable were a load time of less than 30 seconds achieved, however in such a case, information must be provided to the user on the progress of the wait (this was ignored for testing purposes).

A single user was employed to load the web-page on the server using Internet Explorer 7. This negated any delay associated with network topology and traffic. Internet Explorer 7 was previously loaded so as to remove any load time associated with the start-up of the browser. The timer started when the link to the standard was clicked and stopped once the page finished loading as indicated by the status bar. The web page was compiled in release mode so debugging did not affect the timings. Also any graphics and styling were disabled to test just the time required to load the data.

4.7.3 RESULTS AND EXPERIMENTATION

The application was timed at 1 minute 39 seconds to load the subset of the standard that was loaded into the database. This measured performance is much worse than downloading and opening the PDF version of the standard. This initial development of the prototype included no code optimisation and the resulting file was 10Mb. The file size can be used as an indication of how well code optimisation is working and the amount of further effort required. Literature in Section 2.6 suggests at a high performance home page was expected to only be around 30-35Kb (King 2008), with the average page being 312Kb.

As the application was developed using ASP.NET, a typical optimisation is to remove/disable the viewstate used by ASP.NET. The viewstate enables ASP.NET to maintain state information for web-pages; it grows with the number of elements on the page. This optimisation actually produced no noticeable change in file size or speed.

Literature suggests many other strategies for improving the performance of web-pages (King 2001, Yahoo Inc. 2009). The most relevant were selected and implemented. They included:

- Removing white space and characters used to format a document for easier human reading.

- Using default attributes.
- Reducing the number of elements on the page by removing design elements such as divs and spans where possible.
- Using short one character names for attributes, classes, etc.
- Removing any un-required closing tags when using HTML.
- Using fixed table widths for complex tables.

In the initial development, each table cell had rowspan and colspan attributes declared even if they were 1 (the default if not declared). Names and classes were given long descriptors suitable to aid a human reader (e.g. "Num" rather than "n"). Table 4-2 provides the load times observed and the file sizes after several of the listed optimisations.

Table 4-2: Overview of Optimisation Results

	Load Time	**File Size**
Initial Development	1min 39s	10Mb
Remove viewstate	1min 39s	10Mb
Remove white space	1min 38s	9.62Mb
Remove rowspan=1 & colspan=1	1min 35s	9.27Mb
Reduce number of elements (removing divs, spans etc.)	1min 30s	8.31Mb
Use 1 char attribute names	1min 30s	8.26Mb
No closing tags	1min 29s	8.19Mb

Although the document has many large complex tables, fixed width optimisations were not implemented primarily due to the data being dynamically driven from the database. As such, the layout of each table would be difficult to ascertain. In addition, optimisations at this point were not showing significant increases in performance. Having only achieved a load time of 1min 29s after optimising the constructed HTML, it was clear that further HTML optimisation was going to be of little benefit as reducing the file size by 2MB had resulted in only a 10 second saving.

These experiments only assessed performance from the optimisation of the ASP.NET pages. Performance could also be enhanced through Database and Software optimisation. However, the database already had most of the common performance enhancements (Shubho 2009a, Shubho 2009b) such as indices configured on the tables for the queries being executed, use of stored procedures, selects statements returning only the column data required. As such, no performance gain was expected at the database level. Therefore, any further optimisation would be required from the software IIS and SQL Server. It was considered that optimisation at this level would not generate significant saving and was not tested. It was considered that for the MIL-STD-6016C, change would be infrequent and so, in a networked environment, caching would greatly enhance the performance.

Modern web-applications like Facebook resolve the loading of data loading subsets and using AJAX once the user reaches a predetermined distance from the end of the current loaded data to fetch the next subset. This gives the user the feeling of scrolling the whole document. However, if the user wished to scroll quickly, they would still be met by waits for each subset to load. Literature in Section 2.5 suggests that scrolling is not suitable for web based applications with Nielson (Nielsen) strongly against its use.

4.7.4 PAGING

One of the main causes of slowing down the performance was the volume of database queries. A query had to be made to the element table to determine which four possible tables the data was found in. The implementation of the document model could, however, be changed such that the element table held all information across all four types of data (paragraph, table, section, graphic), however, this would result in massive numbers of blank or null fields in the database.

As outlined above, displaying the whole document on a single HTML page would lead to the perception of poor quality due to the page load time. Alternatively a HTML paging solution was required which would reduce the amount of information per web-page. By reducing the transmitted data the web-page will load within acceptable time limits. By implementing a paging system, there is no real functional loss. Document searching is still possible as the search is performed on the database and not the screen, the only loss is the ability to scroll the whole document which, given its size, provides no functional benefit. The MS Word version does not allow scrolling of the whole standard only the current document. It was therefore believed that paging would still produce a usable system and via the enhanced navigation, an improved usability, thus achieving objective A1 (Improve navigation).

The BAE Systems document model (Figure 4-5) did not include the concept of a page. Therefore implementing paging introduces the question of what defines a page and is a page concept required. Should the HTML page represent the same paging boundaries as the PDF document? A page in terms of PDF and MS Word is a fixed dimension based on the assumption the content is to be printed to a physical sheet and as such the amount of context is constrained by these physical dimensions. PDF is fixed when the author publishes the document, however with MS Word, the concept of a page is simply re-evaluated each time the user chooses different page dimensions or the font is changed. In addition, MS Word allows the user to choose between Full Screen Reading, Print Layout and Web Layout, with each view the concept of a page changes and the amount of content differs. Consider devices such as a Kindle, the concept of a page here differs from the printed book, and within itself changes with font size. It is therefore believed that the document should not define a concept of a page and such a concept should be allowed to be flexible.

Within large technical documents there are many logical break points – either at section or sub-section boundaries, or simply on a fixed or user-defined number of elements. It was observed from the MIL-STD-6016C, that sections can be many thousands of pages long, which leads to the same issues in trying to optimise the application for fast load times. Therefore, a fixed number of elements per page was used to generate the load time for a single page. Testing with values of 10 or 20 elements resulted in page loads of less than 10 seconds. Problems did occur when treating a table as an element. Tables in technical documents can be extremely large and as such a page with several large tables took over 10 seconds to load.

4.7.5 EVALUATION

The performance of the ASP.NET application prevents the ability to render the whole document without the wait time exceeding what typical users are prepared to wait. The rendering of the whole document at once is not considered an issue, as it is considered to provide no functional benefit compared to a paging based system. Therefore a paging system can be used to obtain acceptable page load times to maintain a quality interaction with the prototype. Large tables present a challenge as these can degrade performance beyond acceptable wait times. This can be eliminated by displaying a limit on rows, but the header row must then also be reproduced to prevent loss of understanding.

Literature in Section 2.5 has suggested that this may result in slight disorientation and reluctance to adopt if the users are familiar with the existing source. However, given the size of the MIL-STD-6016C, it is believed to be highly unlikely that users will have the memory recall ability. Therefore, it is considered that removing the scroll all document functionality would not prevent adoption. Couple this with the navigational gains and it is believed that losing this ability will provide no negative effect.

What should define a page? The term page is dependent on its context. The PDF version of the standard is fixed as to the amount of content it terms a page. However, this contrasts with MS Word, where the page becomes a flexible object, this flexible approach is also supported by other devices such as the Kindle. The lack of a page concept in the BAE Systems document is, therefore, not seen as a deficiency - in fact including a page concept would introduce one. A page is a physical concept which is context dependent and as such should not be introduced to the document model as a container, as this would remove any flexibility based on changes to the physical output.

4.8 SUMMARY

The existing sources for utilising the MIL-STD-6016C suffer several usability shortfalls. The Microsoft Word version is spread across numerous documents making navigation tedious. The PDF is styled as if it contains hyperlinks, but these are only styling and not real links. In addition, the hierarchical menu in the PDF is limited to three levels; however the whole standard is located in one file, unlike in Word. Therefore, an HTML prototype was developed to alleviate the shortcoming of the PDF and the Word formats. The material is available in a single source, although given the performance this requires a paging system. Hyperlinks are enabled and the hierarchical menu is available to any depth. This improves the navigation (objective A1). Additional new interfaces utilising the semantic model have demonstrated further enhanced navigation which it is believed should also aid understanding (objective A2) by making it easier to negotiate the appendices through the new interfaces.

5 IMPROVING STORAGE AND RISK MIGRATION

5.1 CHAPTER PREFACE

This chapter discusses a change to the meta-model technology following a risk migration exercise, which enabled the review of a new reflective approach to generating dynamic web content directly from the models. A reflective approach means that a single source could be used by both model tool and document rendering agents. This removes any synchronisation or generation of multiple sources, thereby helping achieve objective A3 (Improved Quality Control). The novel, reflective approach is presented and its performance evaluated. The chapter is based on the included Paper 2 (See Appendix B) and also contains material from 2010 additional paper (Clowes et al. 2010).

5.2 INTRODUCTION

Chapter 4 concluded that the application inherited from BAE Systems' early work suffered from poor performance. However, this poor performance was deemed acceptable given the improved usability of the material. It was assumed that much of the performance degradation was related to the object-oriented structure of the models being maintained in a relational database. Could the models have been stored in an alternative method that might alleviate this?

In addition, the approach in chapter 4 required multiple copies of the standard to be maintained, such that they could be used across the differing applications. For example, model comparison, document rendering, code generation etc. This multiple source file approach introduces potential quality control issues as the sources need to be kept synchronised. It also increases any maintenance costs as not only does each source need to be maintained, the business must also maintain staff with the correct skillset to perform this maintenance. A solution which alleviates this, by facilitating a single source method of storage would reduce costs for the business and improve quality control.

During the project an ideal opportunity was presented to make this shift. The underlying technology utilised by BAE Systems ceased development and support, requiring a risk migration strategy. This chapter reviews the approach in applying an alternative storage method for:

a. Performing a risk migration as part of the overall BAE Systems research project.

b. Reduce the required storage mediums to help achieve objective A3 (Improved Quality Control).

c. Review the performance for rendering the standard.

5.3 RISK MIGRATION AND IMPROVEMENT

Full scale development and support of the XMF tool ended in 2008. In early 2009 the business focus of the research shifted into investigating a successor as part of a risk reduction exercise investigating candidate successor tools for XMF. This exercise and subsequent transformation of meta-models was performed under this project at the request of the sponsor.

The chosen successor was the Eclipse Modelling Framework (EMF). EMF is an open source project built within the Eclipse framework (The Eclipse Foundation 2009a). It is defined as:

> *"EMF is a modelling framework and code generation facility for building tools and other applications based on a structured data model. From a model specification described in XMI [XML Metadata Interchange], EMF provides tools and runtime support to produce a set of Java classes for the model, a set of adapter classes that enable viewing and command-based editing of the model, and a basic editor. Models can be specified using annotated Java, XML documents, or modelling tools like Rational Rose, then imported into EMF."* (The Eclipse Foundation 2009b)

Eclipse Modelling Framework (EMF) provides automated meta-model generation from an XML schema, as much of the source data is available in XML generated via bespoke BAE

Systems parsers for XMF, migration of the Data Dictionary and Message Catalogue proved to be feasible. EMF provided the meta-model and base application; additional projects were required to facilitate model management functions. There were several options for performing this task. Epsilon (See: 5.4.2.1) was chosen as this only required knowledge of one language (Epsilon Object Language (EOL)) to perform all of the model management tasks required at BAE Systems, thus reducing the required Skillset that BAE Systems would need to maintain. Some of the functions/languages provided by Epsilon were very similar to the languages provided by XMF (e.g. XMap≈Epsilon Transformation Language (ETL)), albeit using differing syntax.

Meta-models were converted from XMF to EMF by hand. This was due to the slight differences discovered between the two technologies. The main difference was the containment relationship applied by EMF. XMF did not utilise or require any containment relationship. Containment defined where within the object graph the data would be held. EMF allows for the model to be defined without containment relationships, however, using this approach the only way to populate and work with the model is via the native Java interface provided with EMF. The models based on this approach would not be usable with the Eclipse Workbench. The approach was also advised against by Epsilon experts.

This presented two methods for implementing containment, a data-centric approach or a logical approach. By inspecting XMF data files, XMF implicitly utilised a logical approach. As EMF allowed the choice, which approach was best for the BAE Systems' models? A data-centric approach contains elements at a root level. Any relationships within the model are then simply references to the element contained under the root. A logical approach has the containment on the relationship; therefore objects can be contained in a nested structure.

Considering the document model, a data-centric approach would store all paragraphs at the root level, and then the document is simply defined by the order of the references. If another version of the standard is released, the paragraphs defined by the previous version would not need to be duplicated unless they had been changed in this version. A logical approach would contain all elements under the version object, therefore, if the paragraph had not been changed, it would still require being duplicated. The comparison is summarised in Table 5-1.

Table 5-1: Comparison Data Centric Vs. Logical Containment

	Data Storage Centric	**Logical Model Centric**
Ease of navigation of models	The data can be navigated using either the storage relationship or the semantic relationship to the same effect. All objects are stored at the root node. Therefore objects can be accessed either directly from the root node or by navigation through the parent child nodes.	The storage relationship is the semantic relationship. Therefore the navigation is no more or less complex than the data storage approach. All children must be navigated to from their parent.
Ability to utilise Eclipse Workbench/ Complexity of populating models	All objects are available to create at the root level, there is no restriction on creating objects under parent objects. The process is two stage, first create the objects, then declare the relationship.	All objects that can exist under the parent object can be created. It guides the user through what is allowed and as a result of the model structure presents the information in a more logical manner.
Amount of modification to XMF models	New relationships would need to be created for the storage.	The models would very closely resemble the XMF models.
Amount of duplication of elements	No element would need duplicating as they all exist under the root node.	A new version would require the duplication of all objects. As objects only exist under their parent, any change to a high level object would require all the sub-objects being duplicated.
Readability of models	Lower because models are polluted with implementation details.	Models remain clean – more ‘logical’.

Each approach has its own positives and negatives which means either was an acceptable solution. MIL-STD-6016C is a huge document hence a data-centric approach might have been more suitable, however, it does not change frequently. In addition, disk space is now becoming relatively inexpensive. A logical approach restricts the type of data being created, aiding people unfamiliar with the models to define them. As such, after discussion with the wider BAE Systems project team a logical approach was utilised for the containment in the EMF models.

5.4 MODEL DRIVEN WEBSITES

The prototype in (Section 4.5.3) utilised a relational database. The database schema was manually constructed and was becoming increasingly more complicated and difficult to construct as the semantic models evolved to include multiple inheritance. In addition, the data needed to be maintained in multiple sources such that the model engine and ASP.NET rendering could be used. Following the risk migration exercise the models were then stored as XML Metadata Interchange (XMI) and were no longer being utilised in the proprietary software of XMF, hence the full suite of Eclipse tools and projects could be utilised easily.

Section 2.4 has identified that using a relational database is not ideally suited to the complex object structure of the semantic models. However, there is the possibility of ORM, ORDBMS. As the risk migration has resulted in the utilisation of EMF, the Teneo project (Taal & Irawan 2012) enables the EMF models to be held in a database via an ORM technique rather than as XMI. Therefore, using this technique would remove much of the complexity from the earlier approach of manually constructing the database model and the transformations. Using Teneo reduces the need to transform automatically manages the transform of the XMI models into a RDBMS.

Also, in chapter 2, the concept of model driven websites was reviewed. To work within the business constraint R2 (Only use approved software), the project aimed to utilise web-based rendering, and the data was available within the model. All approaches in 2.7 utilised a generative approach to producing the HTML output. This resulted in static content that would need to be maintained and constantly synchronised. In addition, the approaches would require several additional models being created and maintained. Although an existing model driven website approach automates the process of generating the HTML, this must still be executed each time the models change.

An aim of the research was to improve the quality control of the material (A3), achieving this becomes more difficult when the number of sources that need to be maintained increase. Although Teneo results in the XMI model being stored in an RDBMS and this could subsequently be used by the ASP.NET application, the manner in which it is stored is structured around EMF which makes utilising it directly complex. A reflective approach would reduce all of this, and one which could be reflective over an EMF model means that there would be no requirement to maintain data in another source. Epsilon provided the Epsilon Generation Language (EGL) which could achieve a generative approach to producing the static HTML. However, unlike XMF Mosaic, Eclipse and Epsilon are open source projects with the ability to be extended and modified.

5.4.1 PROPOSAL

One option was to use JSP in order to produce dynamic HTML pages from the EMF models. Since EMF is a Java-based library, this capability was available out-of-the-box. However, there was a major disadvantage to this. If JSP were to be used for this purpose, it would be necessary to either generate Java code from the Ecore meta-model, resulting in a generative step once more, or navigate the models using the cumbersome reflective syntax of EMF.

The BAE Systems project had excellent working relations with the University of York Computer Science Department who are experts and lead developers in the Epsilon framework. The team at York were approached with this concept. Webservers like IIS and Tomcat are typically extensible to allow for new server-side languages. Dimitris Kolovos from York, advised that a serverlet could be developed that locates requested EGL files and instigates their execution, returning the response back to the webserver. The team at York subsequently developed this serverlet, such that the approach could be evaluated.

5.4.2 INFRASTRUCTURE

This section is taken from Paper 2 (Appendix B). It provides a technical overview of the technologies used.

5.4.2.1 Epsilon and EGL

Epsilon is a component of the Eclipse Modelling GMT project that provides tools and domain-specific languages for Model-Driven Engineering. Epsilon comprises a number of integrated model management languages, based upon a common infrastructure, for performing tasks such as model transformation, comparison, merging, in-place transformation, inter/intra-model consistency checking, and model to text transformation. All languages in Epsilon build on the Epsilon Object Language (EOL), an OCL-based imperative model navigation and modification language, and can be used to manage models expressed in different technologies such as EMF, MDR and XML.

Epsilon Object Language (EOL). EOL - the core language of Epsilon – combines the procedural style of scripting languages such as JavaScript with the declarative style of OCL for querying and filtering collections. EOL is a mature language that boasts a wide range of features (Kolovos et al. 2006) such as support for managing multiple models of arbitrary modelling technologies in the context of the same program, tight integration with Java enabling developers to instantiate Java objects and call their methods from EOL, support for defining operations in the context of existing types, reuse facilities for defining and importing libraries of operations, support for user-interactions and support for transactional management of models (where the underlying modelling technologies provides such capabilities).

Epsilon Generation Language (EGL). EGL is a template-based language that targets model-to-text transformation (Rose et al. 2008). EGL adopts a syntax that closely resembles server-side scripting languages such as JSP and PHP. An EGL template consists of two types of regions. Dynamic regions (enclosed within [% %]) contain executable statements and expressions, while static regions contain plain text that is output verbatim. For example, consider the simple EGL template in Figure 5-1.

```
[%for (i : Integer in Sequence{1..3}){%]
Number [%=i%]
[%}%]
```

Figure 5-1: Example EGL Template

EGL is a preprocessed language; EGL templates are transformed to EOL programs (in a similar manner to the way JSP pages are transformed to Java servlets) which are then executed in order to produce the output. By building on top of EOL, EGL inherits the rich set of features that EOL provides and which were outlined above. In addition, EGL provides a range of task-specific features such as support for dynamic template instantiation and invocation, and support for mixing manually written code with generated code.

While EGL was originally developed to support code generation, its modular design makes it possible to use it to produce text in non-file output streams as well. In principle any model-to-text transformation language with similar characteristics such as XPand, MOFScript or the OMG M2T could have been used instead.

5.4.2.2 Tomcat Integration

Tomcat is an industrial strength, Java-based web server with built-in support for the JSP server-side scripting language. Therefore, the first option was to use JSP in order to produce dynamic HTML pages from the EMF models. Since EMF is a Java-based library, this capability was available out-of-the-box. However, there was a major disadvantage to this. If JSP were used for this purpose, there would be a need to either generate Java code from the Ecore meta-model or navigate the models using the cumbersome reflective syntax of EMF. By contrast, integrating Tomcat with EGL would allow the use of the concise, closure-based syntax provided by EGL (Rose et al. 2008) to query models, without needing to generate and deploy code for the respective Ecore meta-models in Java.

Although Tomcat comes with built-in support for JSP, like the majority of web servers, it also provides a flexible model for integrating additional server-side languages. This is achieved through the URL mapping mechanism which allows developers to map request URLs to custom servlets. In the case of EGL, an EGL servlet was implemented which was responsible for serving calls to EGL pages and mapped it to requests which ended with .egl as shown in Figure 5-2.

```
<servlet>
<servlet-name>egl</servlet-name>
<servlet-class>org.eclipse.epsilon.egl.servlet.EglServlet
</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>egl</servlet-name>
<url-pattern>*.egl</url-pattern>
</servlet-mapping>
```

Figure 5-2: Tomcat EGL Servlet mapping

Once the EGL servlet was invoked as a result of a client (browser) request, it was responsible for locating the respective EGL template for each request, executing it and returning the produced text to the client. Similar to JSP pages, EGL templates can access several built-in variables such as the *request* variable which allows a template to retrieve information related to the particular request (e.g. parameters), the *session* variable which allows templates to query and set session-wide properties (e.g. for authentication), and the *response*, *config* and *application* variables. These variables are inherited directly by the Java servlet specification (Sun Microsystems 2009). To interact with EMF models, each EGL template is provided with a shared instance of the *ModelManager* class which provides operations for loading, storing and disposing of EMF models (Epsilon Eclipse Modeling GMT component 2009).

5.4.3 PROOF OF CONCEPT

Using the Tomcat/EGL integration, three different templates were developed in an alpha test application. These templates covered three distinct areas of the Link 16 TDL modelling work. The areas covered were the Data Dictionary, Message Catalogue and the prose document. The prose document template regenerated a rendering of a subset of the standard in the same view style as the PDF format. This template was designed to test scalability as it utilised over 900,000 instances of classes.

The Data Dictionary and Message Catalogue were intended to allow users to traverse the hierarchical structures and follow any cross-references between them. Navigation was provided by using hyperlinks and specifying new parameters using the HTML GET method. Providing parameters enabled the template to restrict the data and traverse the hierarchy. An example of this is shown in Figure 5-3. This example returns all child instances conforming to a specific type (InitialWord, ContinuationWord or ExtensionWord), as defined by the passed parameter (wordType). The EOL select operation returns a set where all instances conform to the type and the instances id matches the additional parameter (wordID) that is supplied. The excerpt also corresponds to the meta-model excerpt seen in Figure 4-4.

```
var wordType := request.getParameter('jword1');
if(wordType = 'I')
{x := y.contains.select(t|t.isTypeOf(InitialWord)).first();}
else if (wordType = 'C')
{x := y.contains.select(t|t.isTypeOf(ContinuationWord) and
                            t.id = wordID.asInteger()).first();}
else if (wordType = 'E')
{x := y.contains.select(t|t.isTypeOf(ExtensionWord) and
                            t.id = wordID.asInteger()).first();}
```

Figure 5-3: Excerpt of selecting element defined by passed parameters

Having determined the set or instance to display, a custom print operation is called to render the data. The custom print operations (e.g. x.print();), produce the HTML code for displaying the instance attributes to the screen. In Figure 5-4, a short excerpt is given for rendering the top level class JMessage of the Message Catalogue. This results in a HTML table displaying the data associated to a JMessage.

```
operation JMessage print() {%]
<table>
<tr><td>Name:</td><td>[%=self.name%]</td></tr>
<tr><td>Family:</td><td>[%=self.family%]</td></tr>
<tr><td>ID:</td><td>[%=self.id%]</td></tr>
...
```

Figure 5-4: Excerpt of a custom print operation

It was envisaged that engineers should be able to utilise the modelled data to produce required regulatory material according to subsets of the standard. To achieve this, the solution was required to be able to create and/or edit the data. Figure 5-5 demonstrates the ability to select and edit the data of a JMessage.

```
var editme := c.contains.select(x|x.id = msgID and x.family =
msgFamily).first();
if(editme.size() == 1){
editme.name = msgName;
editme.family = msgFamily;
editme.message = msgID;
myCatalogue.store();
}
```

Figure 5-5: Example of editing a message

5.4.4 EVALUATION

This approach has demonstrated that reflective model-driven web engineering is possible. It improves on generative approaches by reducing the amount of duplicated data and therefore reducing the risks and problems associated with synchronisation. With no synchronisation this helps to achieve objective A3 (Improve quality control), by providing improvements to benefits B2 (Prevent inconsistency) and B4 (Update more effectively).

By using the Epsilon platform, a range of model management functions can be developed using the common syntax provided by EOL, which, allows developers to reuse code across the different Epsilon languages. Reuse of code is not possible in most generative approaches, as their model management functions utilise differing languages with no common syntax. EOL is also beneficial in comparison to OCL as it combines the procedural style of scripting languages such as JavaScript with the declarative style of OCL for querying and filtering collections.

Although EOL is a new language, the syntax is similar to JavaScript. In addition, engineers with no modelling experience are not required to learn multiple languages for model management functions such as, transformation languages like QVT or ATL or constraint languages like OCL. For these reasons, the learning of EOL is not considered an issue; it could even be seen as a benefit. Also, no technical knowledge of dynamic web scripts or languages are required, this reduces the knowledge required by a modelling engineer to produce a dynamic web-based application.

However, the approach does shift away from model-driven engineering principles slightly. Most generative approaches utilise a model and subsequent transformations to generate the dynamic script. The approach currently ignores this model driven approach to the development of the interfaces in favour of a programmatic style that utilises the model driven functions available through the Epsilon platform.

5.5 ASSESSING PERFORMANCE

The approach of utilising the EMF model in a reflective manor was successful, however, as the approach was novel there was no knowledge of its performance and hence suitability for use by BAE Systems. This section assesses the performance of the approach.

5.5.1 EQUIPMENT

The experiments were carried out in a standalone environment to rule out potential performance issues with the creation of a network. They were run on a Toshiba Satellite Pro laptop running an Intel Core 2 Duo T7300 CPU running at 2.00GHz. The machine had 3GB of RAM and the Hard Disk Drive was a Toshiba MK1637GSX 2.5-inch 160GB running at 5400rpm. The operating system was Windows XP Service Pack 3. This was the same hardware used in the performance test for the ASP.NET prototype in section 4.7.

Apache Tomcat version 6.0.20 was the webserver being run using port 8080. The Epsilon version used was the latest compiled source files as of the 28th October 2009, this includes libraries for EMF version 2.5.0.v200906151043. Non-automated testing used Firefox 3.5.5 as the web browser and automated testing used Apache JMeter 2.3.4. Apache JMeter is a Java desktop application which is designed to load test functional behaviour and measure performance. The Java Virtual Machine used was 1.6.0 16 from Sun Microsystems Inc.

5.5.2 TEST PLANS

The 3 main experiments conducted to assess the performance and suitability of this solution for the potential deployment to TDL engineers at BAE Systems were:

- Basic 'CRUD' (Create, Read, Update and Delete) tests
- Simple Page Load Testing
- Realistic Stress Testing

As an additional experiment the performance of the solution for large scale models was also evaluated by reviewing the load time of a model containing a large number of elements. A large scale model was considered to contain over 500,000 elements. 500,000 elements were considerably more than the number of elements in the Message Catalogue or Data Dictionary, but not as large as the number of elements captured in BAE Systems document model (900,000) at the time.

5.5.3 RESULTS

5.5.3.1 CRUD Testing

CRUD or Create, Read, Update and Delete testing is intended to demonstrate the very basic functionality of persistent storage requirements. Within web applications BREAD or Browse, Read, Edit, Add and Delete is seen as more appropriate. BREAD extends CRUD by emphasising the requirement to list or browse collections of data. BREAD testing was performed manually on both the JMessage viewer and the Data Dictionary viewer. The aim of BREAD testing was to demonstrate that Tomcat and EGLServlet could enable a single user to directly perform the BREAD operations on an EMF model via a web browser. The tests involved simply checking that an index page could display a list of elements and these elements could be read individually, the data for an element could be edited, and an element could be created and deleted. Manual testing of this functionality showed that browse was achieved through the index page and that the individual elements could be read, edited, deleted and added as needed. Validation of these tests was performed visually ensuring that any test resulting in modification to a model was reflected in the stored EMF model on the hard drive after the server was terminated.

5.5.3.2 Simple Page Load Testing

If the solution could enable BREAD operations, could this be performed in an acceptable time period for a user? Section 2.6.2 indicated that a user is only prepared to wait a few seconds with the worst case being a limit of 30 seconds waiting.

Simple page load tests aimed at evaluating the performance of a single user accessing and navigating the prototype tool were conducted. As these pages were dynamically created, the file size fluctuated with the amount of data being displayed. Table 5-2 provides some exemplar file size information, it highlights the file size changes a user would expect whilst navigating the application. By comparing the data it can be seen that the jmessage.egl page was considerably smaller than the data dictionary index page. The actual index pages consist of 20 links for the jmessage and 1573 links for the data dictionary.

Table 5-2: File Size Samples

Message Catalogue		Data Dictionary	
Page	**File Size (KB)**	**Page**	**File Size (KB)**
Index	1.49	Index	217.97
J2	2.78	DFI 272	1.8
J12	3.38	272|007	1.44
J12.6	2.06	Data Item	0.71
J12.6C6	2.5	DFI 799	10.28
J12 Field	0.72	799|017	1.12
J17	0.98	799|096	1.13

To establish the performance, the index page was used for testing, being one of the larger outputs. Table 5-3 provides the data for a single user making a request for the jmessage.egl index page. As can be observed, the maximum load time was experienced at the start when the model is first loaded into Tomcat memory. The more requests made the lower the average load time and standard deviation became.

Table 5-3: Load Test for jmessage.egl Index Page

Requests	Load Time (ms)			
	Min	**Max**	**Avg**	**σ**
100	38	4337	84	427.41
250	37	4337	58	271.23
500	37	4337	48	192.01
750	37	4337	45	156.84
1000	37	4337	44	135.86

Table 5-4 shows load times for the data dictionary index page. As the file was larger, the load times were longer, an average of just under 3 seconds. Once again, after the initial load the average and standard deviation reduce as the model now sits in Tomcat memory.

The results of Table 5-3 and Table 5-4, only considered a single user connecting to the server at one time. As a web served application, it needed to be able to handle multiple users. The data dictionary file size was near to the file size of 315KB, considered to be the average file size of a webpage in 2008 (King 2008). As such this file was used in testing in favour of the JMessage viewer.

Table 5-4: Load Test for data_dictionary.egl Index Page

Requests	Load Time (ms)			
	Min	Max	Avg	σ
100	2709	6777	2761	403.60
250	2709	6777	2744	255.73
500	2707	6777	2737	181.44
750	2706	6777	2735	148.60
1000	2706	6777	2735	128.99

Tomcat, by default, limits the number of concurrent connections from the local machine to 200. Therefore, using JMeter, the application was stressed by attempting to make 200 unique requests for the data dictionary index ramped up uniformly over 1 minute. Each user looped their request 10 times ensuring that all 200 users were still active at once. The result is shown in Table 5-5.

Table 5-5: Load Test for 200 Users in 1 Minute (loop: 10)

Requests	Load Time (ms)				Failure Rate (%)
	Min	Max	Avg	σ	
2000	72	263113	223488	62318.05	6.35

Multiple connections degraded system performance rapidly. The application was even unable to return a valid web page for 6.35% of requests. On analysis of the failures (there were 127 in total) 115 of these were the first access requests. This was due to the model not being loaded into memory. If the model load is ignored then only 12 failures occurred, which is a more respectable failure rate of 0.6%. However the average load time had increased from under 3 seconds for a single user to just under 223 seconds for multiple users.

In reality, the prototype tool is not likely to experience 200 user requests a minute; and approximately 10 requests per minute was more representative. To establish a suitable number of concurrent users, the test was run again varying the number of users connecting per minute. For this test the model was pre-loaded into memory to remove the effects of this activity from the observations. The gap between connecting users was still uniform and non-representative of real world connections. Table 5-6 shows the results.

Table 5-6: Load Test (loop: 10) Model Pre-Loaded

Users per Min	Requests	Load Time (ms)				Failure Rate (%)
		Min	Max	Avg	σ	
200	2000	534	271122	222896	26787.51	0.1
100	1000	276	123821	103299	7870.76	0.1
50	500	5892	52181	46016	9463.98	0
25	250	2840	24208	19358	5205.36	0
10	100	3391	11768	9586	1836.45	0

The failure rate in each case was acceptable. However, even at 50 user requests per minute, the average load time for the index page was 46 seconds. This was far from acceptable. With 10 users the load time was an acceptable 9 seconds. It should be noted that these tests were conducted with each user looping their action 10 times. For each user the next request was started immediately after a result was returned from their previous request.

5.5.3.3 Realistic Stress Testing

The simple load tests used uniformed data which was non-representative of the real world. The tests also focused on accessing just the single index page. More realistic test cases were developed to produce data representative of real world use. These test plans accessed the Data

Dictionary index page before randomly selecting from 8 sequences of events. These sequences represented a user selecting a link and navigating several steps through the data dictionary. Some sequences followed links to the JMessage page as well, others linked to the start of another of the sequences. These sequences were looped between 1 and 5 times to represent the user spending time navigating the structures. The rate at which a new user started the test plan was controlled by a uniform random timer which added a level of randomness to the rate at which the users made requests over the minute. An example of the results is shown in Table 5-7.

Table 5-7 shows the index page for the data dictionary and a sample of other pages that were requested. The index page managed to maintain a low failure rate. The failure rates also reflected the randomness of the test, with higher failure rates for lower user requests per minute in some cases. This was believed to have been caused by the loading on the system peaking during that particular request under the weight of the other requests being made. A summary of the tests are provided in Table 5-8. The summary demonstrates that the prototype in a realistic application can achieve acceptable results even with 50 user requests per minute. The failure rate was at 0.5% and the average load time of a page was 5 seconds. The standard deviation, however, was 13 seconds with a maximum page load of 70 seconds. Given the results of the 50 user requests per minute the test was conducted with 100 users which can also be seen in the Table. The performance has degraded to unacceptable levels for this number of users as the average load time was 11 seconds with a standard deviation of 27 seconds. The failure rate had also climbed over 0.5% to 0.8%.

Table 5-7: Example Realistic Stress Test Results (Users per Minute)

Page	Users	Requests	Load Time (ms)				Avg File Size (B)	Failure Rate (%)
			Min	Max	Avg	σ		
Index	10	100	71	13058	10367	2366.373	220995	1.0
	25	250	4229	30262	22837	6609.444	223200	0.0
	50	500	144	69979	50239	15437.879	222318	0.4
1953|001 DI 7	10	15	71	124	90	16.068	334	0.0
	25	31	79	172	114	27.455	412	3.2
	50	58	75	522	170	59.034	376	1.7
268|008	10	48	102	343	250	50.555	1657	2.1
	25	120	94	736	422	110.198	1652	1.7
	50	252	153	1744	859	211.960	1652	1.6
804|001 SUB 5	10	26	1611	7428	4654	1569.957	15601	0.0
	25	76	1577	16931	11363	4109.071	15601	0.0
	50	112	2582	39428	25825	10760.306	15601	0.0
J.2.4.I	10	21	142	225	225	40.519	2928	0.0
	25	70	97	670	418	133.489	2925	1.4
	50	120	155	1458	845	273.210	2928	0.0

Table 5-8: Summary Realistic Stress Test Results (Users per Minute)

Users	Requests	Load Time (ms)				Avg File Size (B)	Failure Rate (%)
		Min	Max	Avg	σ		
10	1565	45	13058	1110	2650.502	16362	0.3
25	3711	55	30262	2649	6217.821	17489	0.3
50	805	52	69979	5041	13194.568	15922	0.5
100	15426	77	130983	11184	27084.301	16715	0.8

5.5.3.4 Large Models

The previous experiments utilised models with only a few thousand elements. BAE Systems anticipated the solution would be required to work with much larger models containing over 500,000 elements. The subset of the document model that had been imported as data was approximately 900,000 elements. This was considerably more than the number of elements in the Message Catalogue or Data Dictionary. Another prototype application developed was the TDL document viewer. This prototype used the meta-model of the document structure. Examples of the types of classes used in the meta-model are cell, paragraph and section.

Early tests encountered problems with memory when using models of this size. The Java heap space allocated to Tomcat must be increased if using large models. In this instance the heap was increased to 1 GB. No automated testing was conducted on this prototype. This was because a manual test to load the whole model in one webpage was timed at approximately 15 minutes. This included the time to load the model into memory. The load time was reduced if the model was already in memory. However, it still took minutes to execute. Running the EGL file through Eclipse and not using Tomcat resulted in a time of approximately 3 minutes.

On investigation, this load time delay was found to have been caused by the workings of Java return statements. The performance overhead was a result of how the Epsilon Language is designed and mainly how Java utilises the Stack to evaluate the return of a function/method.

Epsilon made heavy use of type test functions to verify correct types. If the code of the application is modified and the use of these type tests functions removed, the prototype can loop through the model and output basic HTML in less than 10 seconds. However, to achieve this BAE Systems would have to run and maintain a customised version of the Epsilon source. It also resulted in the EGL files utilising no functions, both which hindered the objective A3 (Improved quality control).

This indicated that a paging system was needed for this prototype to enable acceptable performance. The previous application using ASP.NET and Microsoft SQL Server also required a paging system. A complete load of the models using that technique took approximately 1 minute 30 seconds. Should improvements be found to the return statement issue then it is quite possible that the EGL and Tomcat solution can rival or better the previous ASP.NET application for these large models.

5.5.4 EVALUATION

The simple prototypes developed have demonstrated that the core functionality of a web page can be achieved using this solution. Browse, Read, Edit, Add and Delete (BREAD) operations can easily be written and developed using this approach. The solution does experience a delay as the models are first loaded into the Tomcat memory. As part of the configuration of the server, an action could be to load the model into memory. However, the Java garbage collector may remove the model if it has not been used for long periods of time and this may result in users experiencing some start up delay. This is not considered a significant issue in the context of the prototype.

Stress tests focused on one individual page demonstrated that the load time performance is only acceptable for between 10-25 users. A more realistic stress test mimicking typical usage indicates that the prototype was able to handle between 25-50 requests per minute. This is somewhat in excess of what is expected to be required for deployment of the tool to the TDL engineers' desktop within BAE Systems.

The prototype for testing of a large model experienced extreme performance problems. However, this appears to have been caused by the prototype's use of operations which return results. A simple model containing large amounts of data could perform well provided the application used few operations and simply returned the data stored in the model. Should any improvement be found for the performance of the return functionality then these experiments should be re-evaluated as a small improvement in this area could result in a large performance gain overall.

All these tests have been performed using Tomcat in its default setting with the exception of increased heap size. Performance enhancements to Tomcat (Chopra et al. 2007) such as using non-default HTTP connectors like the Apache Portable Runtime (APR) or New I/O (NIO) HTTP connectors may result in improved performance. In addition, the server was left in development mode such that context could be auto-deployed. Turning this setting off is expected to enhance performance marginally. The Java Virtual Machine (JVM) was used in its default configuration. This means that the client VM was being executed, the JVM could be configured to run in server mode which could enhance performance slightly as well.

5.6 SUMMARY

This chapter discussed the risk migration undertaken at BAE Systems following termination of XMF Mosaic support and development. The shift to the Eclipse framework presented the opportunity to improve the quality control (objective A3 (Improve quality control)) through the utilisation of a single source for both model functions and document rendering. To achieve this, a reflective approach to generating dynamic HTML directly from an EMF model using Epsilon Generation Language was proposed. The performance of this approach, being novel, has also been assessed and shown to be comparable to that achieved by the ASP.NET prototype. Although the process did not appear suitable for request intensive applications, it is suitable for BAE Systems needs.

6 HYBRID MODELS

CHAPTER PREFACE

This chapter reviews the information contained within the MIL-STD-6016C document and following analysis proposes five relationships as enablers for the project's aim of hybrid semantic-document models. The chapter is based on the published article in the Computer Standards and Interfaces Journal (Paper 1. See Appendix A).

6.1 INTRODUCTION

Tactical Data Link 16 is one standard that it is believed would benefit from the use of hybrid semantic-document models. Given the wealth of material available and due to the close association between the researchers and the study, the approach was an action research (Baskerville & Wood-Harper 1996) based "single-case" case study (Benbasat et al. 1987) method, focusing on the MIL-STD-6016C. Given the volume of MIL-STD-6016C, a single-case design should have been sufficient to allow the construct and proof of the hybrid semantic-document models concept. It is also believed that the framework developed by analysing the MIL-STD-6016C can thus be generalised such that it can be applied not just to STANAGs or Military Standards, but to other domains where the capture of complex rigorous information is maintained in a prose based document.

6.2 SAMPLE SELECTION

Previous extraction and modelling of the semantics at BAE Systems had led to a level of knowledge of the document. As such, a selection of pages from the standard was selected using stratified sampling, key areas of the material were grouped into strata, and a random sample selected from these strata that represented the semantic model and/ or the document structure. The strata were not equivalent in volume, but were intended to cover a cross section of the differing document structures and the semantics that would need to be mapped. For security and sensitivity reasons actual headings and text used in this chapter have been substituted with fictional material that maintains the essence of the original.

Three pages from each strata were selected to form the sample. If the selected page contained a full page graphic, this was ignored and an alternative random page selected. Consecutive pages were also prevented. The random sampling method was a non-probabilistic judgement by a member of the BAE Systems research team who had knowledge of both the semantic models and the document sections and content. The samples obtained are shown in Table 6-1.

Table 6-1. Sample Selection

Strata	Pages
Payload definitions	B-38, B-146, B-824
Message definitions	5.1-15, 5.1-440, 5.2-12
Message rules	F-6, K-14, P-53
Transmit and receive tables	5.4-J7.0-6, 5.4-J12.6-1, 5.5-J14.2-1
Constraint information	4-23, 4-29, 4-169
General information	1-17, 3-56, 4-130
Message Uses	5.5-J12.0-15, 5.5-J0.3-1, 5.5-J2.5-2
Minimum Specification	W-18, W-27, W-92

On analysis of the samples, it became apparent that, for the message rules, too much meaning and information was lost by reading a single page. The sample selection for message rules was, therefore, changed to a trace. A non-probabilistic random rule was used and all pages necessary to trace an understanding were selected. For example, Rule F.1.2 covers 2 pages, however, through its references it requires further knowledge of 2 rules, a transmit table and indirect knowledge of 4 other messages. These indirect references were ignored as their exclusion would not prevent basic understanding. This resulted in a total of 6 pages for analysis for this rule.

6.3 LOCATION OF INFORMATION

In maintaining two sources of information and formatting, the primary location and relationships between sources needs to be considered. The semantic model is the obvious choice as the primary information location as the document model contains multiple references to the same object. This information can be maintained singularly in the semantic model, resulting in a change to the semantics being reflected at every location in the document. This supports the objective of reduced duplication and hence improving quality. It is also the model which is likely to experience the most computational activity as viewing the document becomes a rendering activity.

However, is it that simple? The two models can be applied to a matrix to aid the analysis. This two-by-two matrix is represented in Figure 6-1.

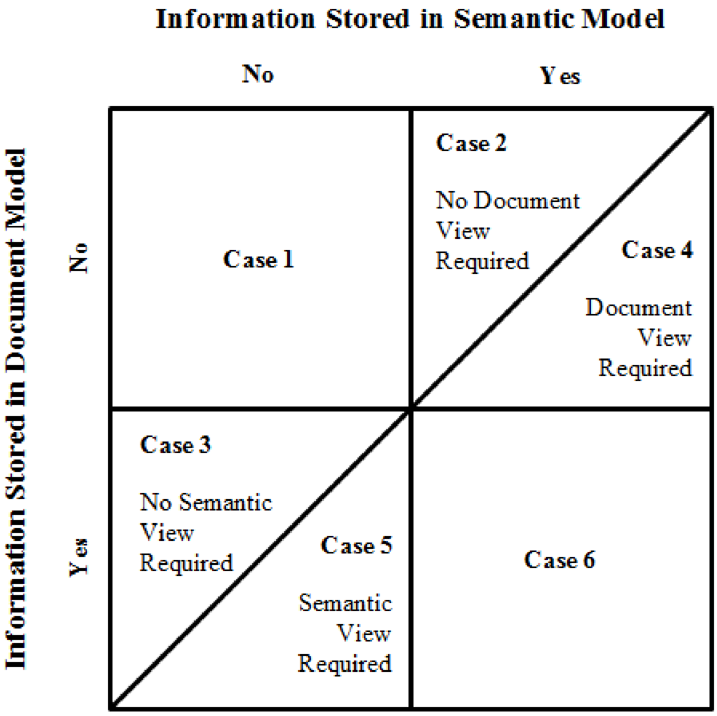

Information Stored in Semantic Model
No
Yes
Information Stored in Document Model
No
Yes
Case 1
Case 2
No Document View Required
Case 4
Document View Required
Case 3
No Semantic View Required
Case 5
Semantic View Required
Case 6

Figure 6-1. Information Store Matrix

Cases 1-3 of the matrix can be ignored. These cases have an impact on the location of information as, in case 1, the information does not exist and, in case 2 and 3, information is not required in the alternative form. In theory, cases 4 and 5 are equivalent, in that the information can be located in either form and then a link or view of the information could be created and maintained in the other. In these scenarios, case 4 is considered the better option because, as mentioned previously, the semantic information will only be stored in one place in the model. The only occasion where case 5 is not a limitation is a singular occurrence (only ever used once in the document model), in which case this can be transferred into case 4 by relocating the information into the semantic model. This would improve any future development as well.

As case 4 represents the situation where the data can be directly mapped from the semantics to the document view or at least derived from the semantic content, case 6 presents potentially the most challenging situation. It represents the case where the same information is represented in two mutually exclusive formats such that one format cannot logically be derived from the other. As such, this case presents the need to maintain a relationship between the two equivalent sets in both sources, this would therefore result in duplicated data across both locations. It would not be possible to maintain the information in a single primary location and still fulfil the requirements of both of both models.

6.4 TDL SEMANTICS

To perform an analysis of the material, knowledge of the semantic model that supports the document prose must also be established. It is beyond the scope of this thesis to present the exact models used, but an overview is provided in this section.

At the lowest level of granularity, there exists a Data Dictionary identifying the set of data types defined for use on the link, these types are identified by a unique key, the Data Field Identifier (DFI) and the Data Use Identifier (DUI) pair, referred to as the DFI/DUI. The set of messages (referred to as J-Messages) that may be transmitted over the link are defined in the form of a Message Catalogue. Messages are functionally-oriented and contain a number of words, referred to as J-Words, each of which contains a number of fields, the type of which is defined by reference to the relevant item in the Data Dictionary (the DFI/DUI).

Hence, Link 16 messages are tree-structured and must conform to certain well-formed constraints, e.g. all bits in each J-Word must be associated with a DFI/DUI (i.e. all fields must have a defined type). Such constraints have been captured in the models and are described elsewhere (Holmes et al. 2007). There are a small number of different types of word, and certain elements of the payload are mandated by the word type. This structural set of information covers a large proportion of the standard, covering approximately 4000 pages or some 53% of the document.

In addition, structural modelling has also been undertaken on other sections which detail the network management and Time Division Multiple Access (TDMA) architecture. Once again, there was much information which could be captured and represented by a meta-model. Another large section of the standard covers the behavioural aspects of the standard with approximately 2000 pages covering the definition of transmit and receive tables and appendix data defining the behaviour and rules related to message transmission, receipt, storage and display. The appendix rules follow an event (or stimulus) - condition - action type structure. There is a clear process flow within which can be modelled.

6.5 SAMPLE ANALYSIS

Through the analysis of the samples, one key association was observed at an extremely high occurrence rate. It almost exclusively forms entire sections of the samples, specifically the message and data definitions. This association is between an element of text and an instantiated class attribute. In many cases the attribute value of the class is presented directly in the document in the form and format defined in the semantic class. To demonstrate this association consider a semantic model for *"Message"* defined in Figure 6-2. Within the document, tables simply render all of a Message's parts, in a table e.g. Figure 6-3.

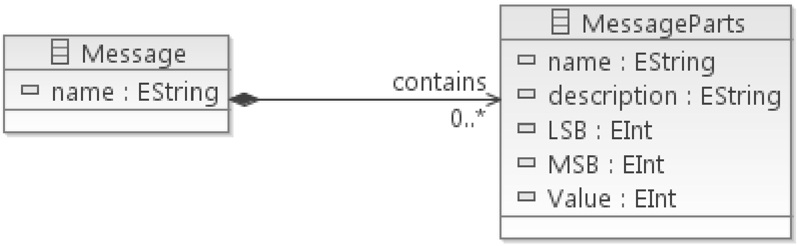
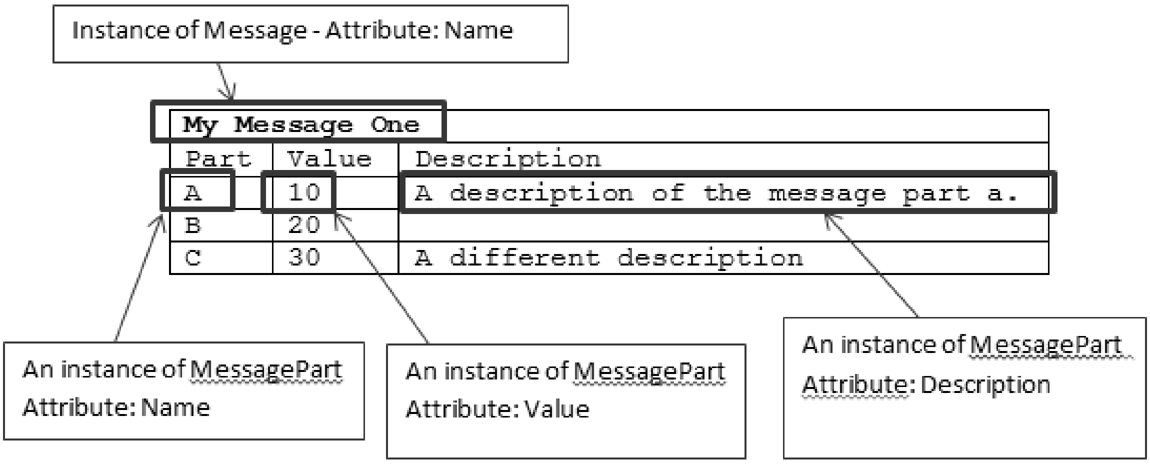


Figure 6-2. Message Semantic Model



My Message One

Part	Value	Description
A	10	A description of the message part a.
B	20	
C	30	A different description

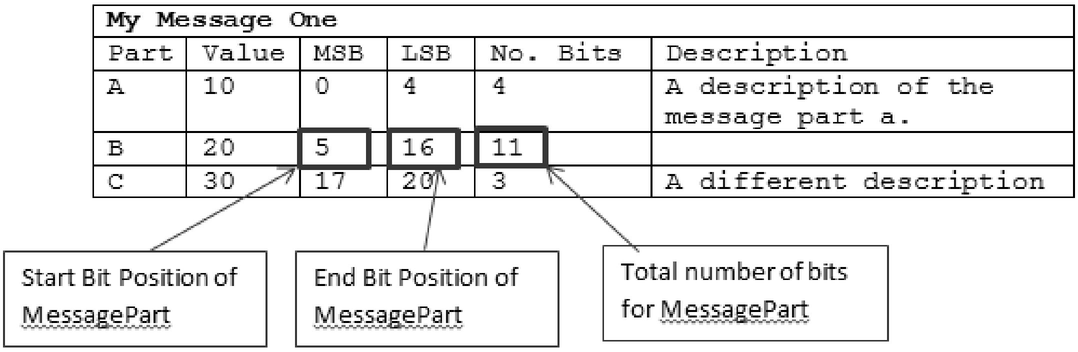
Instance of Message - Attribute: Name
An instance of MessagePart
Attribute: Name
An instance of MessagePart
Attribute: Value
An instance of MessagePart
Attribute: Description

Figure 6-3. Example Document Content (Message)

To facilitate the aims of hybrid models, a relationship would need to be specified such that this observed association can be modelled. As such a *"direct"* relationship should be defined where an instantiated class attribute can be directly utilised from within the document model.

This attribute based substitution can also be observed in other forms. It was hypothesised that the document would contain Boolean information, and the storage format and display format of Boolean data could be different. Most computer based systems use the bit as the storage medium using 0 for false and 1 for true. The hypothesis is that there may exist situations where using a direct replacement of Boolean data would result in illiterate sentence/word structures. Therefore, the bit value should be converted to a more meaningful string for display in the prose, such as True and False or Yes and No. This could be extended to the consideration of numeric display. In some instances the string can be used, but it is the actual numeric value which is stored in the model. Similar synonym based actions can be observed across the samples. For example, the minimum specification defines a set of characters that are used to indicate mandatory and optional settings. The transmit and receive tables also use characters for the source and message use display type. Therefore, Boolean/numeric conversion can be considered as part of the larger concept of synonym substitution, whereby the attribute value is substituted by a synonym. As such a *"synonym"* relationship should be defined to allow for this reformatting.

A more concealed association was observed in the message and data definitions. Within the samples it could be seen that the text could be derived from the semantic class attributes. Extending the example shown for direct data demonstrates this type of occurrence. For this the start and end bits of the message part have been introduced. The new table Figure 6-4, needs to display the start, end and total bits. Using the semantic model, the number of bits can be derived from the start and end bits. Similarly if it had been modelled with the start and length, the end bit could be derived. As such a *"derived"* relationship should be defined where the data displayed in the document is a calculation between attributes on the instantiated class.



My Message One					
Part	Value	MSB	LSB	No. Bits	Description
A	10	0	4	4	A description of the message part a.
B	20	5	16	11	
C	30	17	20	3	A different description

Start Bit Position of MessagePart
End Bit Position of MessagePart
Total number of bits for MessagePart

Figure 6-4. Example Document Usage Semantic Derived Data

The case study document makes use of its highly structured nature to infer further associations. Large sections of narrative are broken into a number of paragraphs. The first paragraph might make reference to a semantic class through the previously discussed "direct" relationship. In doing so there is a clear link established between the paragraph and the semantic data. Any child paragraphs can then also infer the relationship through applying the hierarchical structures of a document model. However, the subsequent paragraph, at a sibling level, can in some observed instances, have an implied association with the semantic class referenced in the previous paragraph. The paragraph makes no direct mention or use of data from the semantic class but, through the terminology, spatial proximity and grammar, a human reader can infer the association. Considering an objective of document sub-setting or semantic searching, these sibling paragraphs would be ignored as no relationship would exist. Therefore, some form of *"inferred"* relationship is required to handle these instances. During the analysis a common misleading association was observed multiple times where the document references a table identifier in relation to discussing the semantic class. Because the document model table reference makes use of the semantic attributes from the reference text, it originally was mistaken to be an association to the semantic model. On closer inspection, it is, in fact, a traditional reference within the document model and no hybrid link needs to be established. What it does raise is the association a table has with semantic data. This highlights again the observations of the inferred relationship as the semantic association could possibly be inferred through its child cells or an association would need to be established as per the inferred relationship.

At present, the technique for modelling the behaviour within the standard has not yet been implemented at BAE Systems. It is anticipated that further relationships may need to be developed to integrate the behavioural semantics. In particular, it is anticipated that the behaviour may not be directly representable in the document prose. In such cases the observed, inferred relationship may be sufficient to handle most cases of behaviour. The main anticipated exception shown in initial behavioural studies implies a need to handle collection associations. Collections are anticipated to handle the situation where a relationship needs to refer to more than one instance of a class. It should be bi-directional in that either a text part is associated with many behavioural objects, or many logically separated text parts define a semantic behavioural object. As such, there should be a "Collection" relationship defined to handle this anticipated requirement.

Therefore, analysis of the sample section led to the identification of five possible basic relationship types:

- Direct Attribute Substitution
- Synonym Attribute Substitution
- Derived Attribute Substitution
- Inferred Relationships
- Collection Relationships

6.6 PROPOSED RELATIONSHIP TYPES

Having identified the required relationship types through the analysis of the sample data, the next stage was to propose the relationship definitions to facilitate this. Throughout this process, consideration was given to applying the definition to the material observed in the sampling. A business decision at BAE Systems, meant that the technology for any model would be the Eclipse Modelling Framework. Therefore, all examples shown and some evaluated proposals are closely coupled to the Eclipse Modelling Framework and the underlying model format Ecore. The Epsilon (Kolovos et al. 2006) meta-model management family tools were also the favoured technique for model manipulation at BAE Systems.

This section discusses the definitions and the considerations made during application to the observed material. It should be noted that in the examples shown, a class name in italics implies the class is abstract.

6.6.1 DIRECT RELATIONSHIPS

Direct attribute substitution appears sufficient to cover most occurrences. A direct substitution would simply require replacing the document text with the value of a semantic class attribute. This relationship can be defined by a relationship to the class and an identifier for the attribute of the class as seen in Figure 6-5.

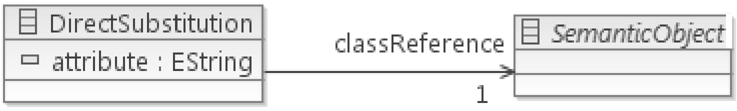


Figure 6-5. Direct Substitution Proposal

However, a question is raised when applied to the identification of the Message Catalogue and Data Dictionary identifiers. These identifiers appear significantly throughout the semantic information within the samples. For example, DFI/DUIs are identified by the pairing of the DFI id and DUI id to form a 7 to 8 character identifier, the J-Messages and their subparts are identified by the hierarchical combining of their ids. The J-Message or J-Word precisely demonstrates the conceptual problem. Is the identifier for a J-Word, a singular reference, or should the J-Word be a composition of references where the parent classes are also referenced by the selective substring of the J-Word identifier? To highlight this, Figure 6-6 shows the example of how the J-Word Z9.3X2 could be composed and represented in the document.

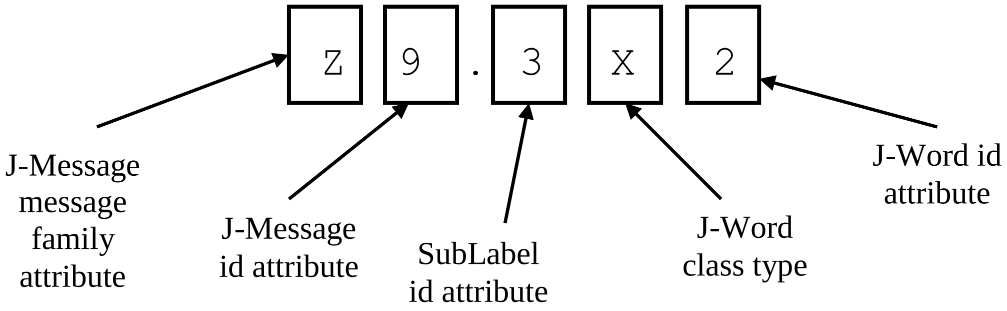


Figure 6-6. Decomposition of J-Word ID

When referring to the J-Word, the full string reference is used e.g. “Z9.3X2”. Using simply “X2” would be ambiguous as there are many X2 J-Words of other J-Messages. The critical question is whether instances within the prose are made up of the composition of parts shown in Figure 6-6, or the full string itself should be the only reference to generate the prose?

By creating all instances within the document as compositions of the hierarchical parts, there is a clear and direct link to each level within the hierarchy. This makes it simple for discovering abstract queries such as finding paragraphs related to high level hierarchical objects. However, ignoring the current use of the military standard, considering a scenario whereby the document is being generated through a series of iterations, this composition approach is extremely poor at handling change. For example, consider an iterative step whereby it is decided to move a J-Word X2 or SubLabel Z9.3 to Z9.4 as it was entered incorrectly or due to reorganisation. The user would also need to find all instances within the document of the composite relationships as the reference to the SubLabel would not change and would still be valid. To handle this scenario, it would be better to ensure each hierarchical identifier is a singular reference. By using the object model, relations to the parent hierarchical classes could still be discovered, but the document would not contain any direct knowledge.

To facilitate the singular reference, the relationship needs to have knowledge and ability to traverse the meta-model. This traversal is not guaranteed to be fixed depth and the string composition could require differing attributes at each level. Therefore, a programmatic method would be best to generate the necessary text string for the relationship. This method-based substitution relationship, defined in Figure 6-7, should exist in addition to the direct substitution relationship.

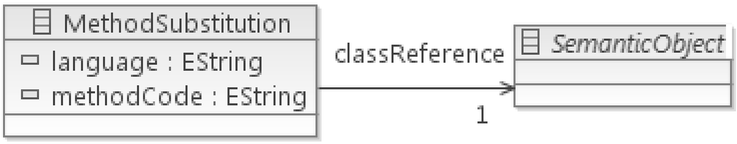


Figure 6-7. Method Substitution Proposal

Decomposition highlights another potential relationship. Should the referencing be seen as the composition of attributes, then the J-Word type, (‘Initial’, ‘Extension’ or ‘Continuation’) is identified by the inheritance within semantic class as shown in Figure 6-8**,** and not by an attribute of the class. Therefore, a relationship to the class type would be required with the same functionality of synonym such that some text could be used to represent the class type.

Although this type of relationship is evident in assessing the composition of the J-Message identifier, there is no other evidence of this elsewhere within the samples. It is, therefore, not deemed necessary to implement this type of relationship. Should it be found that this relationship is required, the functionality could be achieved through the use of the method substitution relationship identified in Figure 6-7.

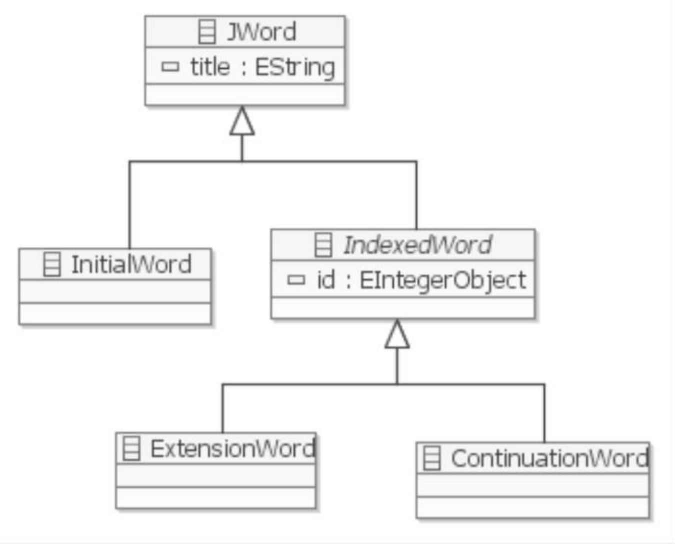


Figure 6-8. J-Word Inheritance

6.6.2 DERIVED RELATIONSHIPS

The observed samples demonstrate a relatively simple case of derived data where the result is the output of an operation between two attributes. This type of simple derivation can be achieved through a predefined set of operations and links to the necessary attributes. Although this type of implementation would facilitate the examples seen in the samples, it is not highly flexible and is fixed to the set of operations that would be defined in the meta-model.

Having established the need for a programmatic method substitution to handle the generation of identifiers in the Analysis Section, this method substitution gives the full flexibility offered by the use of a programmatic language whilst providng all the necessary functionality to achieve a derived substitution. It is proposed, therefore, to drop the derived substitution relationship in favour of using the method substitution relationship. In using the Eclipse Modelling Framework with Epsilon it is proposed that this method substitution be implemented using two techniques. Eclipse has the ability to define eOperations for implementing methods on the class instances. Technique one would be to establish the relation to the result of the defined eOperation. Technique two abstracts out of the Eclipse framework and provides the relation as a text string of code and an implementation language. This would allow us to implement Epsilon as the method language to generate the result. The proposed relationships are defined in Figure 6-9.

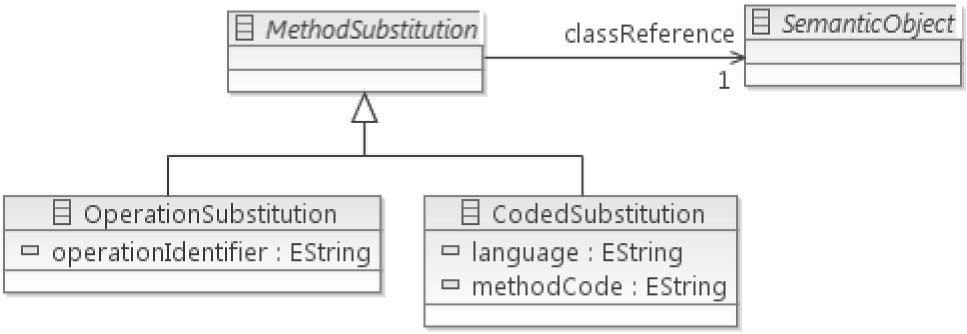


Figure 6-9. Ecore Modified Method Substitution Proposal

6.6.3 SYNONYM RELATIONSHIPS

The Synonym relationship is intended to facilitate the situations where the data held in an instanced semantic class requires formatting into another form. There are 3 potential methods to achieve this.

1. Basic fixed string replacement.
2. Mapped replacement.
3. Method replacement.

A basic string replacement method would allow the instances of the relation to be assigned a fixed string to be substituted, determined as and when the relation instance is created or modified. Although the basic fixed replacement provides a simple solution for existing documents, similar to discussion of hierarchical identifiers, using a basic fixed replacement is not flexible. Working on the same scenario whereby the document may be created or modified in iterative steps, a fixed replacement suffers from the inability to respond to a change to the semantic attribute value. It would, therefore, be possible to use the method substitution relationship proposed in the derived relationships. This gives complete flexibility but requires a programmatic solution. An alternative would be to use mapped replacement. A mapped replacement provides a greater flexibility, when defining the relationship; the user must also define a map between values and the required substitution. This method can facilitate the basic fixed method but provides the flexibility to respond to changes to the instantiated semantic class.

The mapped replacement strategy is not immune to faults, consider the scenario where the document author wishes to preserve the language used in the document such that numeric values are represented by their string name e.g. 3 = ‘three’. The map necessary to handle this scenario would need to be infinite to support possible numbers. Although this scenario could be more easily supported through the method substitution, it is also an extreme scenario used solely to highlight the issue. As the mapped approach also enables the map to be defined within the model, this approach is favoured. With the method substitution required for other requirements, this too can be used for any complex mappings. Figure 6-10 shows the proposed substitution relationship.

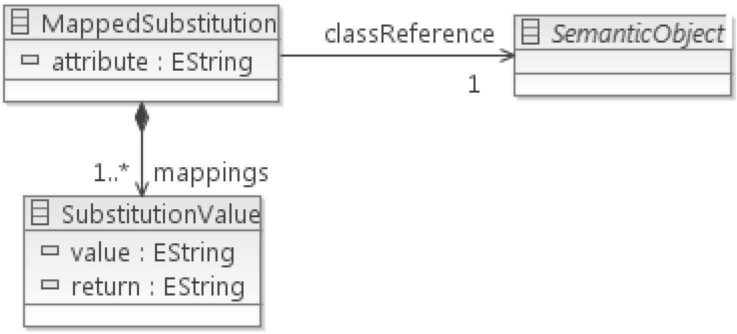


Figure 6-10. Mapped Substitution Proposal

6.6.4 INFERRED RELATIONSHIPS AND COLLECTIONS

The inferred relationship type has no display requirement as it is not designed to be used by a document model to render information. It simply implies an association. As such, the inferred relationship is simply a class with a reference to the semantic class.

Having data returning methods in collection relationships would lead to ambiguity in which element is generating what, and potential synchronisation issues would be created should items be removed, i.e. is the generation still valid? Collection relationships would, therefore, only need to collate inferred relationships. Figure 6-11 proposes the inferred relationship and the ability to create collection relationships. It facilitates many-to-many style associations through semantic inference and document inference classes.

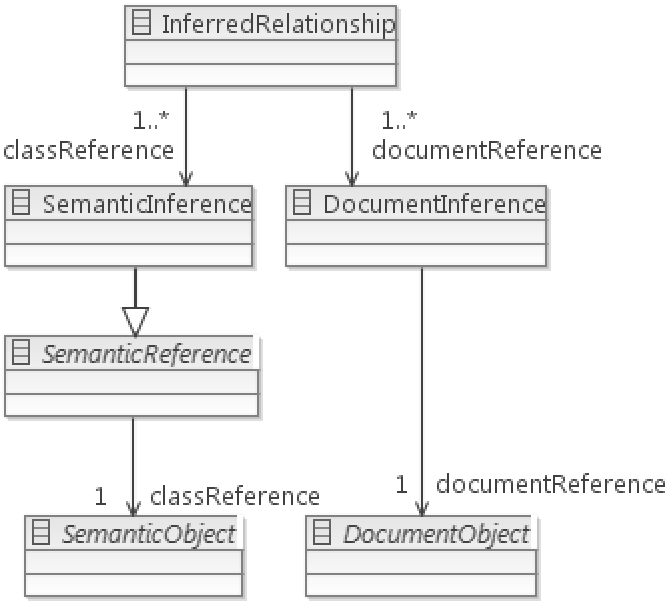


Figure 6-11. Inferred Relationship Proposal

All of the previously defined relationships are inherently associated with a document model object as they are required for display. The inferred relationship is not and, therefore, needs to create this reference to the document. One method would be to define an object/relationship that resides within the document model, or the semantic model. This would require heavy modification and pre-consideration would be required when defining either model, potentially populating the actual essence of the model. This would also go against the computing concept of ‘Separation of Concerns’ (Hürsch & Lopes 1995). By defining a container class such as the Inferred Relationship class in Figure 6-11, this whole concept can be abstracted out into its own supporting model containing just references to the two other models (like a many-to-many table in relational databases). This ensures that the semantic, document and inferred models stay true to their primary concern, semantics, rendering or mapping, and are not polluted by other concerns.

6.6.5 THE SEMANTIC AND DOCUMENT OBJECTS

Throughout the definitions discussed above the concept of an abstract Semantic Object is used as a terminal point in all definitions. A simplistic method of achieving this is proposed through the definition of the class as an abstract inheritable class. To make a semantic class available to be referenced through the proposed definitions, the semantic model would need to be updated such that the class inherits from this new abstract Semantic Object class. This would allow semantic model owners to also restrict access to elements.

Similarly, the Document Object defined for collection relationships can be achieved in two ways, modifying the base document model such that document model objects also have a reference attribute to the abstract semantic class, or defining an abstract document class that document model elements can inherit from. With the construction of an abstract Document Object class that document elements inherit from being favoured.

6.6.6 FINAL PROPOSAL

Analysing the proposed relationships as a group allows for a level of abstraction which aids integration with documents. Mapped substitutions can be seen as an extension of direct substitutions as they both require the class attribute. All semantic substitutions have a requirement to relate to a semantic object such that the final proposal defined in Figure 6-12 defines an abstract class semantic reference which facilitates the abstraction of the semantic association.

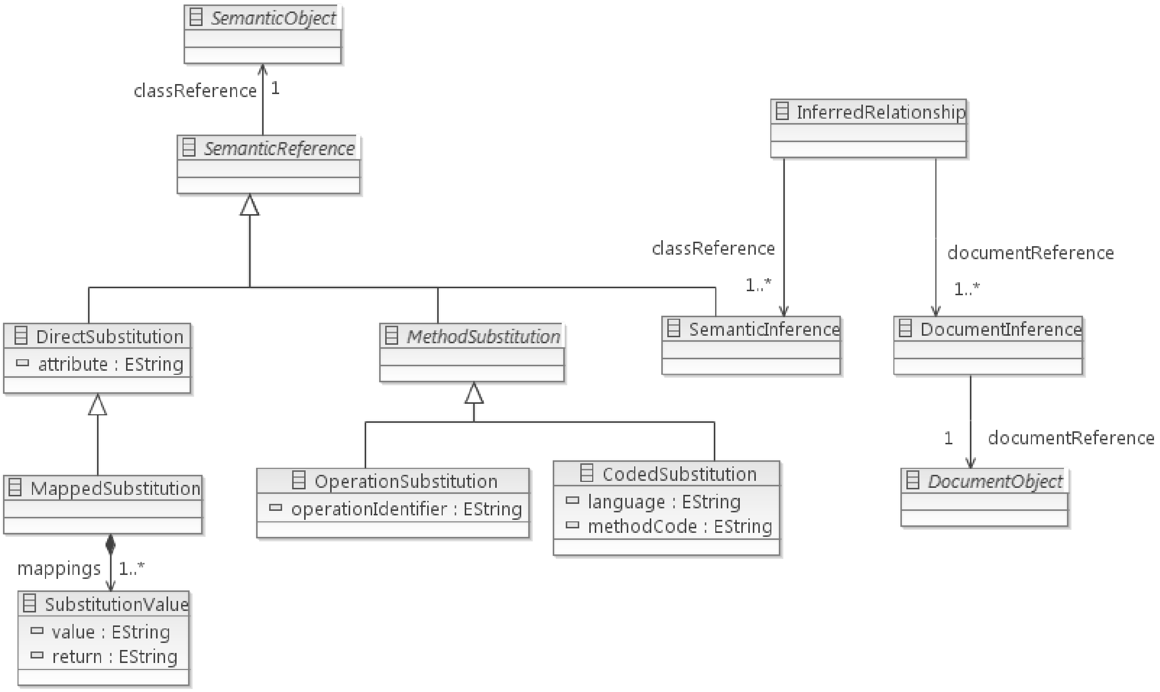


Figure 6-12. Complete Hybrid Linking Meta-Model

6.7 INTEGRATION WITH DOCUMENT MODELS

At BAE Systems, for proof of concept and other legacy reasons beyond the scope of discussion in this chapter, a custom document model was developed to facilitate the capturing and rendering of the military standard. The document model itself can be found in a previous paper (Clowes et al. 2008). With the exception of Inferred Relationships, all other relationships are concerned only at the level of a sub-sentence.

A major design decision on defining these relationships was not to tightly couple their implementation to this BAE Systems proprietary document model. Therefore, to incorporate the relationships, they can be used with any document model that, at least, facilitates the breakdown of sentences into a collection of utterances or text parts. Figure 6-13 demonstrates the integration of the relationship into the BAE Systems model at the level of text part. In performing this integration another abstract class is proposed, Semantic Substitution. This abstract class represents all hybrid relationships that return a text string, such as substitution relationships. The abstract class makes it easier to factor out a semantic inference as being a viable option as a text part.

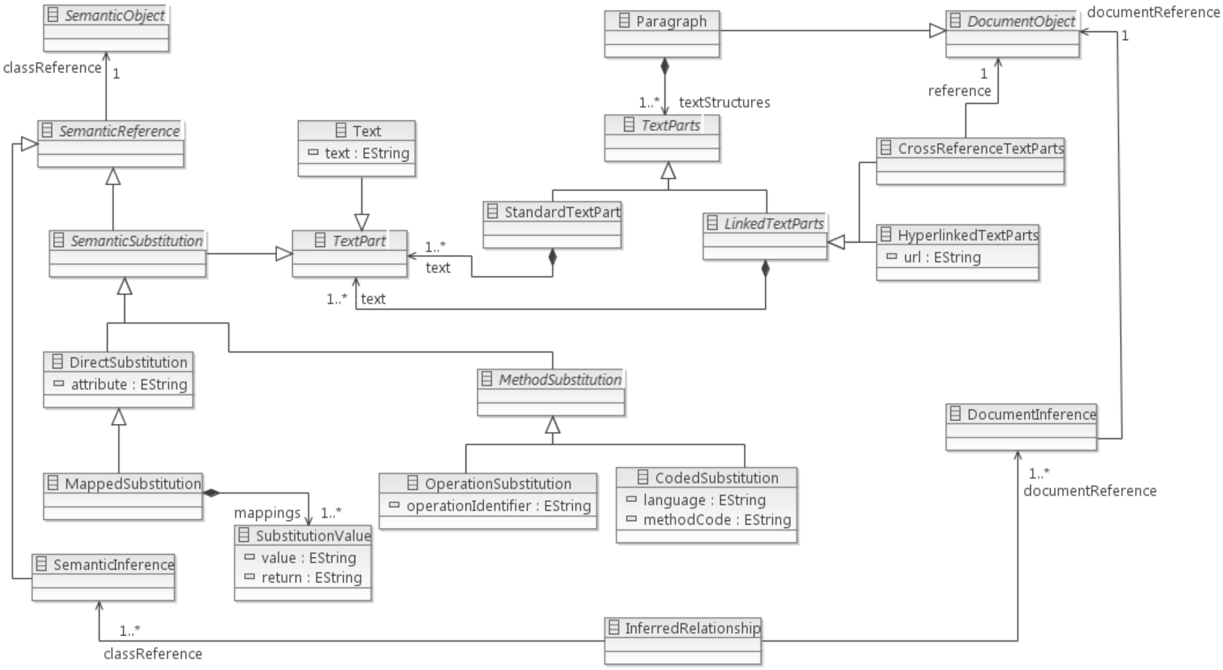


Figure 6-13. Integrated Hybrid Model

With this proposal, the document model itself contains the semantic relationships (excluding Inferred Relationships). Therefore, the document must contain the instances of the semantic relationships. This raises the question as to whether the document should contain the hybrid links. One view would be ‘yes’, as all the semantic substitution objects are simply a construct for generating the rendering, which, in essence, is the primary concern of the document model. The other option would be to define a new document model class which is simply a pointer to semantic substitution instances in a separately maintained, hybrid model. The hybrid linking is now separated from the document model, but the document model still needs an element of change to facilitate the reference.

The first view is favoured; as the purpose of the document model is to be the definition of the rendering pattern for the information. As such, the semantic substitution would fall under this concern and be suitably modelled here under the concept of ‘Separation of Concerns’. The Inferred Relationships which do not define any rendering can be separated out into another model/package.

As it stands, another key factor in the definition of these models is the intention for them to be utilisable as a bolt-on package to existing document models. This has been designed-in through inheritance. As a semantic model would need to be custom developed for each domain/document, extending this area is not considered a disadvantage. The main requirement here would be extending classes to inherit from the Semantic Object class such that they can be referenced. The document is a potential problem for this bolt-on concept. To reference Document Objects, the document class would need to be modified such that document classes inherit from the Document Object abstract class. Within the BAE Systems model this is not a problem as the document model already has a concept "Element", which all referenceable document objects inherit, so the Document Object class is replaced by a reference to the actual document model class "Element".

The Document Object is not the only issue, the Semantic Substitution class also needs to inherit from a document class such that it becomes an option when using the document model. The class would need to have no required fields other than a unique identifier and type. In the BAE Systems model, an abstract class "TextPart" is defined from which Text is a child node. Users can then make use of this abstract class as the inheritable object for the hybrid meta-model. By utilising this abstract class, the semantic substitution becomes available as part of building sentences in the document model. Other document models may not have this level of abstraction. If a text part/utterance is defined as a collection of text objects but the text object is left blank, the Semantic Substitution object could inherit from this, maintaining the idea of a bolt-on package. If this is not possible, then another document model would be required or the existing document would need to be 'tweaked' to inject the required abstraction to allow for null text parts.

If these changes are simple, as in the BAE Systems model, then changes are only required to the hybrid meta-model i.e. replacing Document Object and inheriting Semantic Substitution. In these cases, the meta-model does act as a bolt-on feature where no change is required to the document model. A custom rendering agent would still be required such that the data from these hybrid links can be displayed when trying to render the document.

Therefore, the complete requirements for a compatible document model are:

1. It should, at least, facilitate the breakdown of sentences into a collection of utterances or text parts.
2. It has an object that all document elements are inherited from, or it can be modified such that the document class can inherit from a new abstract class.
3. It has a class at the text part level that does not have any required attributes beyond a unique id.

6.7.1 DITA AND DOCBOOK

DITA and DocBook are both OASIS standards for technical document structure modelling (OASIS 2009, OASIS 2010). DITA, in particular, is highly customisable through the use of specialisations (Hennum 2005). The keyword object of DITA appears to be a suitable point for the proposed Semantic Substitution to inherit. DocBook has a significant number of elements that facilitate the splitting of a paragraph into smaller elements from which to inherit. With regards to the Document Object, it is not clear if this could be achieved through reference to existing elements. Both appear to support the idea of facilitating the hybrid models as a bolt-on package from the perspective of Semantic Substitution but more detailed analysis would be required to assess the integration of the inferred relationships.

6.8 EVALUATION

By using the Case Study method and document analysis research methods proposed in chapter 3 and applying to the MIL_STD-6016C document it has been possible to obtain evidence to support the need from the relationships which form the proposed framework. The case study has demonstrated the need for each relationship type that has been used to construct the framework. This framework has then been evaluated against the project aims and anticipated benefits. The proposed solution does achieve many of the objectives in Section 1.3.1 and the anticipated benefits discussed in Section 1.3.2. Although some of the benefits could be considered achievable through independent models, the novelty of this approach comes from its coupling to the written document. Capturing the interpretation of complex prose, is one such example.

Benefit B1 (Aid understanding of complex information) can be solved through the sole use of a semantic model. Using just a semantic model introduces a second source of information and duplication but, by using the relationships defined, elements of the duplication and hence synchronisation issues are reduced. The main benefit of this approach is through the inferred relationships, as sections of semantics can be clearly linked to sections of prose. This enables novice users of the document to trace a complex paragraph to a formal semantic model. It should help reduce ambiguity through the interpretation of prose by different engineers as there is one formal interpretation underlying it, therefore, achieving objective A2 (Aid understanding of complex information).

The reduction of ambiguity aids the objective of resolving/preventing inconsistencies, benefit B2. By having the relationships, any change to an underlying semantic value is immediately reflected in each instance within the document model. This means changes are not missed by having to go page by page to find reference to the item, hence aiding quality and helping to realise objective A3 (Improve quality control). This one point of change also meets the objective to make updating information easier, benefit B4. Even the inferred relationships helps as, before a change is made, it can be inspected to see where it is utilised within the document and any prose changes can then be made.

It, unfortunately, does not directly improve the situation of resolving missing data (benefit B3), beyond the requirement that the data must exist to be related. The process of generating the required semantic model does aid this, however, in reading the prose to generate the model or starting from a model, it is hoped an engineer would be able to identify any missing elements.

Searching (B6) is also only slightly improved. The semantic model can be searched and, given the relationships, the result can then be returned with links to elements within the document model. This does provide new search methods which works towards objective A1

(Improve navigation), however it can make searching harder. As the data no longer lives directly within the prose, searches from outside of the rendering application are much harder e.g. a system search within files.

As with the searching, the relationships do provide alternative navigation structures through navigating the semantic model and then tracing the relationships back to sections of the document, both of which support benefit B5 (Alternative navigation) and objective A1 (Improve navigation). The biggest potential of this solution is the objective of performing advanced document comparison A4 (Document comparison) and benefit B8 (Document comparison). As a document is supported by an underlying semantic model, the comparison can be made at semantic level rather than the textual level. It is not a benefit in its own right, however, as this comparison could be achieved by maintaining a separate semantic model, but the tight coupling gives the appearance to the user that the comparison is operating on the document.

The final benefit to discuss is the use of model execution facilities (Benefit B7). This benefit is only achievable through using the semantic model and can be achieved with or without the use of hybrid models.

In summary, although many of the anticipated benefits of this approach can be achieved using independent semantic models and documents, this approach benefits by combining them into one. The biggest benefit compared to maintaining independent document and semantic models is reducing duplication and synchronisation issues. It also potentially hides the semantics from users with little modelling background, enabling them to utilise the benefits of models whilst maintaining their user friendly document view but having advanced options, such as semantic comparison, available.

6.9 SUMMARY

This chapter has presented a framework of relationships which allows for the relationship of objects in a semantic model to be represented in a document such that the information can be rendered without the need to maintain the information both in a document form and a semantic modelled form, thus achieving the philosophy aim in Section 1.3. It helps realise all of the objectives in Section 1.3.1, and many of the benefits anticipated in Section 1.3.2.

Although many of the benefits can be achieved through using independent models, a significant improvement of this technique is the close coupling, which reduces duplication and the need for synchronisation.

7 TEMPLATING

CHAPTER PREFACE

This chapter expands on the definition of hybrid semantic-document models through the analysis of the repetitive nature of standards documents. By utilising a template to define and re-use, patterns can be applied to large sections of technical documents improving quality from the perspective of consistency, completeness and efficiency. This is helping to realise further objective A3 (Improve quality control) through only being required to define the template.

7.1 INTRODUCTION

The proposed hybrid semantic-documents meta-models have the ability to provide many benefits identified in Section 1.3. A consideration not evaluated with the proposal of hybrid semantic-document models from Section 6.6.6, is the effect of applying the semantic markup to the storage and communication. Traditionally complex technical standards are huge in volume. Applying semantic markup to these could lead to significant increases in the electronic size of the documents. This increase could result in the storage or communication of the standards becoming prohibitive based on costs or capacities. Any costs would need to be off-set against the benefits obtained through the use of hybrid models. However, an observation during the analysis of the material for the case study domain, TDL link 16 (MIL-STD-6016C), perceived much formatting of semantic-based information needed to be provided in a common layout and style. This leads to the hypothesis that this pattern of reproduction can be utilised, which could also reduce the amount of markup required.

By considering a re-usable pattern, other potential benefits become possible to enhance quality and efficiency through the utilisation of patterns, thus enhancing the ability to achieve objective A3 (Improve quality control). It has been observed that the TDL Link 16 suffers from inconsistency and missing data (Zeigler 2006). A pattern can potentially alleviate these issues. Different writing styles have also been observed during analysis. Applying a pattern could reduce this effect, applying uniformity to the display of information, which tends to aid users understanding (Objective A2).

7.2 THE EFFECT OF HYBRID SEMANTIC-DOCUMENT MODELS

Applying the semantic markup required by the proposed hybrid semantic-document models could lead to significant increases in the amount of space required for the electronic storage of complex technical documents. By using a semantic-document model, even simple text strings and paragraphs require being wrapped in the relevant semantic markup. More complex structures such as paragraphs comprising multiple hybrid links to data in the semantic model will in turn result in significant new markup. To examine the effect and for illustration, a paragraph has been generated in Figure 7-1.

> “Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris ut lectus est. Morbi elementum faucibus turpis, nec vestibulum augue mollis nec. Maecenas in odio justo, iaculis dapibus risus. Duis bibendum consequat dolor, et facilisis lectus vehicula ac. Aliquam erat volutpat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Proin feugiat aliquet condimentum. Nullam semper pretium libero nec fermentum.”

Figure 7-1: Demonstration Paragraph

In its basic form this paragraph is 449 bytes. To apply the document model markup to this paragraph would result in text similar to Figure 7-2. This text now requires 736 bytes for storage. Therefore the larger the paragraph the less effect the markup will have. The MIL-STD-6016C document however, comprises of relatively short paragraphs, this can be seen by the sample study in Table 7-1. This sample, estimates an average characters per paragraph at 78. Assuming the required markup for a paragraph is 287 bytes this would result in an estimated markup increase at 368%. This only considers a basic paragraph of plain text with no utilisation of the hybrid semantic-document model linking objects.

```
<elements xsi:type="DocumentStructure:Paragraph" xmi:id="_Tz89gjMkEeKTY89il_PPBA">
      <textStructures xsi:type="DocumentStructure:StandardTextPart"
xmi:id="_Tz89gzMkEeKTY89il_PPBA">
        <text xsi:type="DocumentStructure:Text" xmi:id="_Tz89hDMkEeKTY89il_PPBA"
text="Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris ut lectus
est. Morbi elementum faucibus turpis, nec vestibulum augue mollis nec. Maecenas in
odio justo, iaculis dapibus risus. Duis bibendum consequat dolor, et facilisis
lectus vehicula ac. Aliquam erat volutpat. Class aptent taciti sociosqu ad litora
torquent per conubia nostra, per inceptos himenaeos. Proin feugiat aliquet
condimentum. Nullam semper pretium libero nec fermentum."/>
      </textStructures>
</elements>
```

Figure 7-2: Example Paragraph Markup

A common technique employed in web development to reduce file size for client side files is minification (Souders 2008). The same principles could be applied to the markup. A particular method used is to substitute method and variable names with a unique minimal character representation. This principle could be applied to the namespace conventions and class and attribute names. Minification would make human readability more difficult but would reduce the file size in the example paragraph to 619 bytes. Therefore a basic paragraph markup would only be 164 bytes with a markup increase at 210% for the MIL-STD-6016C.

Table 7-1: MIL-STD-6016C Sample Characters per Paragraph

Characters	Paragraphs	Characters per Paragraph
24,321	141	173
128,334	2,312	56
3,593,191	45,185	80
1,191	107	11
7,075	284	25

Similar to the size of the paragraph, the effect of the hybrid link object will vary depending on the amount of information. Considering a worst case approach whereby a single character is to be replaced by the new hybrid link, the character would need to be replaced by the link markup. Using a direct substitution link (See 6.6.1) and a Message object of the TDL Link 16 semantic model, the required markup can be seen in Figure 7-3.

```
<text xsi:type="Hybrid:DirectSubstitution" xmi:id="_WR_tQDMkEeKTY89il_PPBA"
attribute="name" reference="./messages.model/#_Tz89hTMkEeKTY89il_PPBA"/>
```

Figure 7-3: Direct Substitution Markup

In this instance the single character is replaced by 148 characters. Obviously, should the semantic datum have been over 148 characters then it could be considered a net saving. However, this is actually a naïve viewpoint as the document would still require the semantic model. The text would still be needed with the additional semantic class structure and markup. It, therefore, could be considered to have a significantly higher byte requirement. The exact requirement would be dependent on the semantic model design. To illustrate, Figure 7-4 provides the markup for the semantic message object referenced in Figure 7-3. This is an additional 374 characters and includes the message container (Catalogue) and its required fields.

```
<Messages:Catalogue xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:JMessages="TDL.Models.Semantics.JMessages"
xmlns:Messages="TDL.Models.Semantics.Messages" name="A Message Catalogue">
  <contains xsi:type="JMessages:JMessage" id="1" family="J" name="J1"
xmi:id="_Tz89hTMkEeKTY89il_PPBA"/>
</Messages:Catalogue>
```

Figure 7-4: Semantic Message Object

Many complex technical documents usually contain large sections of similarly formatted representations of data. In MIL-STD-6016C there is the message catalogue and data dictionary. These sections define the correct data and values and have a common layout for each type of datum. The two sections alone comprise approximately 53% of the standard. Much of this could be defined by a repeated pattern. For example, a single template could be designed for the display of information regarding a message, and a control loop could be specified that iterates over all messages to form the message catalogue within MIL-STD-6016C.

Should a template be used this could reduce amount of space required for the electronic storage. Section 7.5 evaluates this space requirement by comparing against the available formats of the MIL-STD-6016C e.g. Microsoft Word and Adobe PDF.

7.3 QUALITY CONSIDERATIONS

Zeigler (Zeigler) observed that, the MIL-STD-6016C suffers from inconsistency and incompleteness. This was further supported by the BAE Systems Modelling team’s work in defining their semantic model. In places the different writing styles of multiple authors can be noticed. In such large technical based documents it can be extremely difficult to prevent inconsistency and missing data, especially documents developed through multiple iterations. Often, with these prose-based documents, the only method available for checking completeness is through manual proof reading. By defining a template, consistency and accuracy can be automated.

Within MIL-STD-6016C there are many repetitively formatted sections that could benefit from this templating approach. In particular, the message catalogue and data dictionary sections both follow a consistent format for rendering their hierarchical structure. The data dictionary comprises 1027 pages with the message catalogue being 2148 pages. Using the message catalogue as an example, a template can be defined for the rendering of a message, with a loop in this case defined as all messages within the semantic model. The hierarchical nature of these sections also leads to embedded templates. The format for rendering a message within the catalogue is to display all of its child sub-labels and the data elements for these sub-labels. Templating these means that no data is missed, provided that it exists in the semantic model and the loop definition encompasses the data.

A template presents the possibility to improve an author’s efficiency as it rapidly responds to change. Adding more data to the semantic model or modifying the data and the control loop could automatically discovers and renders these changes without any need to modify the document model. Similarly, should extra pieces of data be required to be displayed for all messages, a simple change to the template is all that is required and then the extra data could be viewed by all iterations of the template. These features can dramatically improve the quality of produced documents.

7.4 TEMPLATE DESIGN

To design a templating meta-model that is utilisable within the proposed hybrid semantic-document models, it is first important to identify the requirements for such a template. These requirements are seen as:-

- Ability to contain standard document model elements such as paragraphs, table cells, etc.
- Ability to define a relationship to the semantic model that can be evaluated to return multiple data elements.
- Ability to reference attributes of returned data elements.

Another potential requirement, which is not seen as critical, is the ability to reference a static semantic class as the meta-model defined in 6.6.4 achieves.

7.4.1 TEMPLATE DEFINITION

The largest challenge for the template object is how the hybrid link classes can be instructed to use the template's control loop rather than a fixed reference to semantic class instance. The template object itself must be made available within the document model. This was achieved by extending the propriety document model (Figure 4-5) by having the template inherit from the element class.

7.4.1.1 Duplicated Links

The simplest solution is to duplicate the hybrid link classes such that the duplicates operate within the template. To achieve this, the duplicates need to reference the template they apply to rather than utilise the SemanticReference class. This resolves the issue of whether the currently proposed links (section 6.6.6) have a direct reference to a semantic class defined at design time and would result in a meta-model similar to Figure 7-5.

However, it introduces its own problems in limiting the point in time at which these types of relationships should be made available. The hybrid links need to be accessible as a text part, but they should only be made available when used within a template. Although it would not be possible to create a template based hybrid link without a template existing, as there must exist a template to reference, the option would always be visible. This could become confusing for authors. To alleviate this, the document model classes would need to be duplicated as in Figure 7 6, such that these duplicates operate under a template. The two text-part classes can then be differentiated by allowing access to the relevant hybrid links.

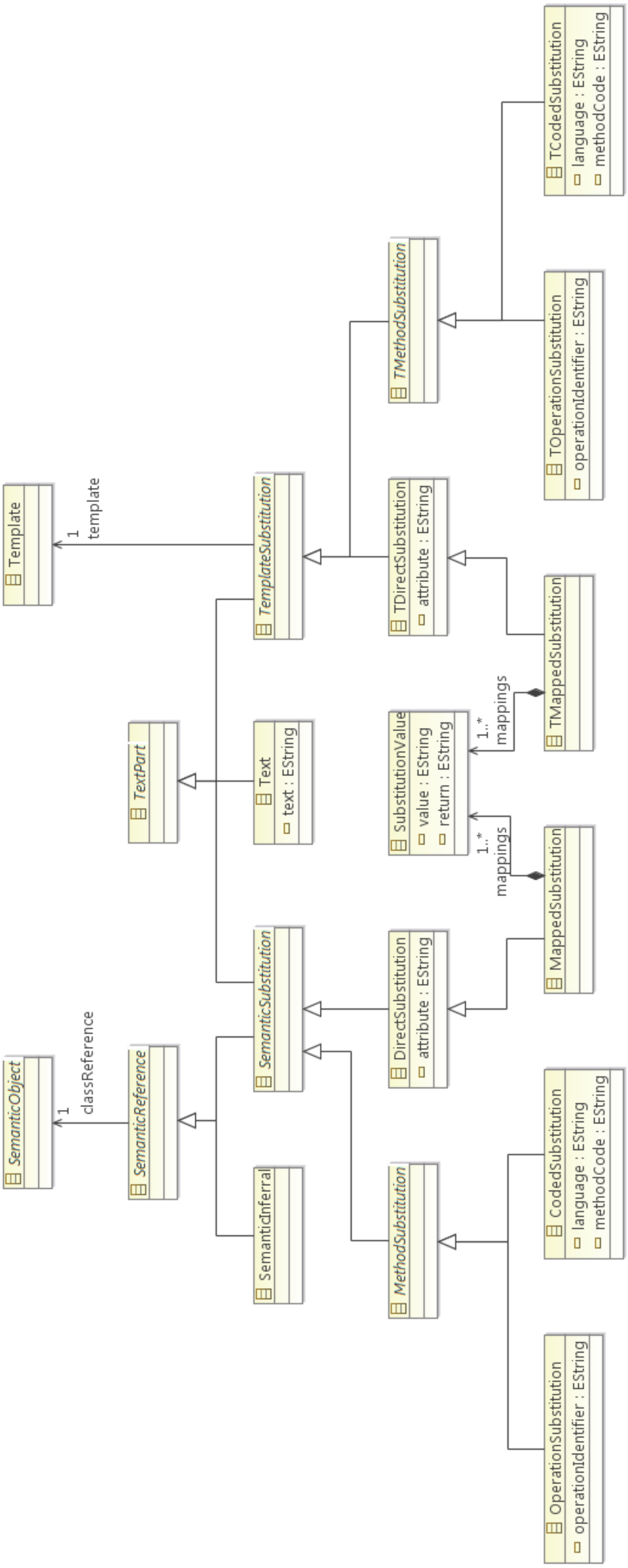


Figure 7-5: Duplicated Hybrid Link Meta-Model

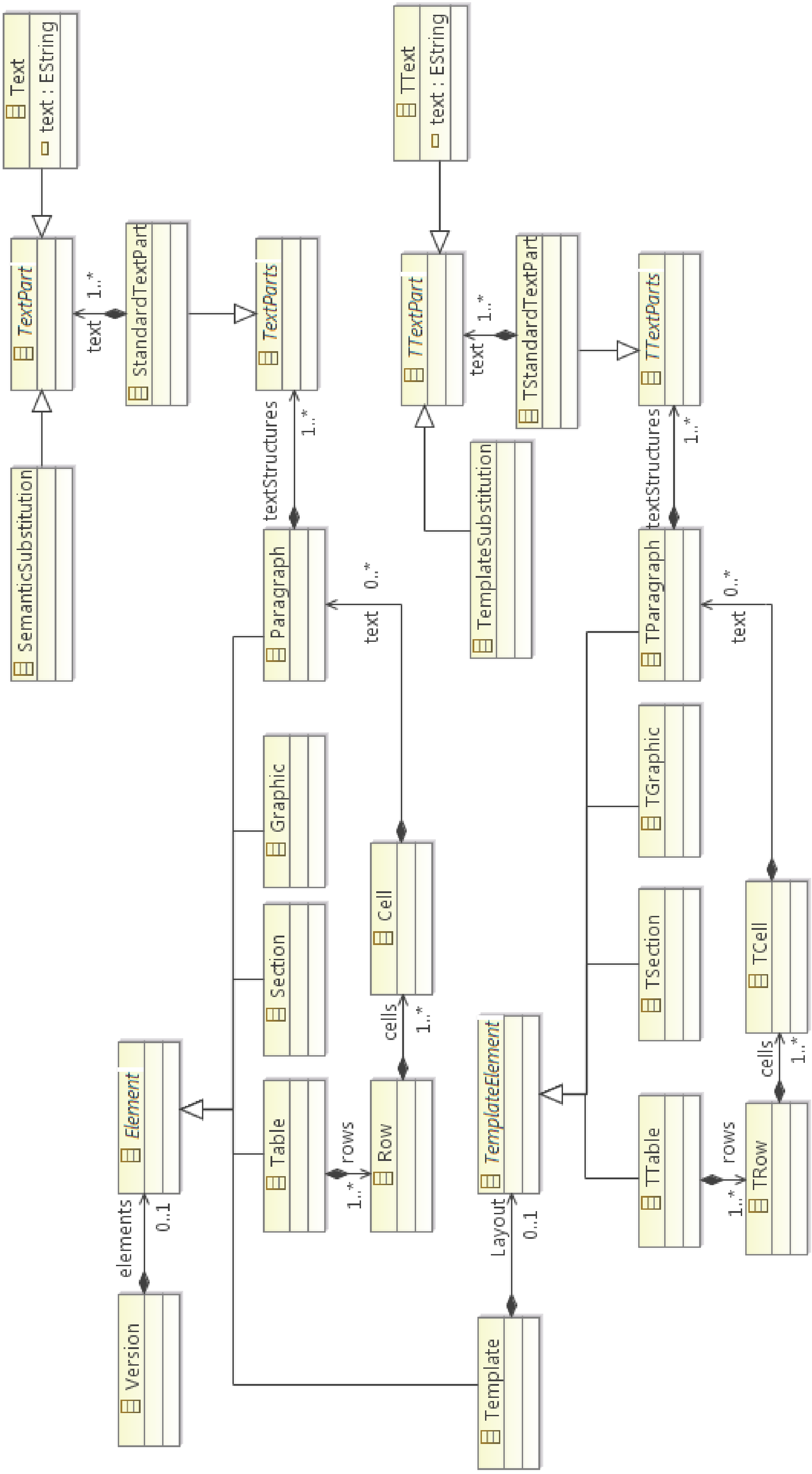


Figure 7-6: Duplicated Document Model Meta-Model

7.4.1.2 Abstracted Relationship Class

The method of duplicating the classes introduces a lot more complexity and issues in maintaining, and further developing the hybrid links. A much better solution would be to be able to utilise the existing hybrid links (section 6.6.6). The problem with using the current links is manipulating the SemanticReference. To achieve this manipulation a meta-model as defined in Figure 7-7 is proposed. By abstracting a new class Relationship, it maintains the existing links whilst facilitating templating. Previously the SemanticReference would have directly related to a SemanticObject, with this new proposal it now references a relationship. The StaticRelationship class allows for the non-template derived hybrid links to be used.

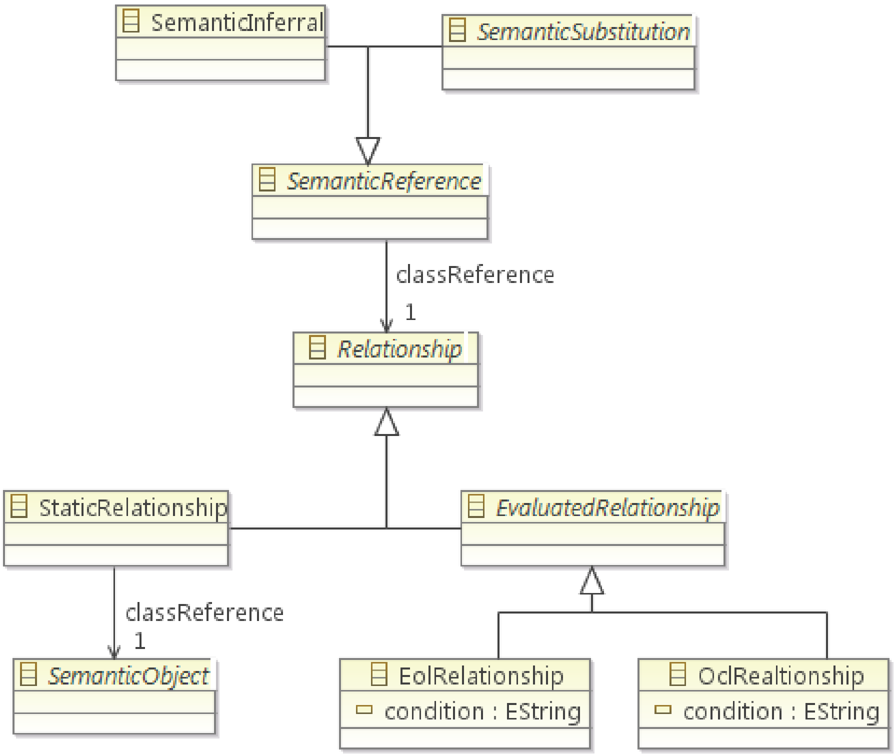


Figure 7-7: New Relationship Class

As an alternative to a static relationship, the abstract EvaluatedRelationship is available. The template will contain the evaluated relationship object and as such will only be available to select should a template have been defined. This new abstract class removes any need for classes to be duplicated, greatly improving maintainability.

7.4.2 CONTROL LOOP

A critical requirement is the ability to define a control structure that indicates all the required classes to iterate over. There are several methods that could be used to achieve this.

7.4.2.1 Modification of Class Reference Relationship

The simplest way to achieve this is to modify the one-to-one class reference relationship of the SemanticReference abstract class into a one-to-many relationship. This allows document authors to define which classes the template should be iterated over when defining the template. It is not, however, responsive to change. If the author were to add a new class instance to the semantic model, the template would need to be modified to include a reference to it. This would therefore not aid the missing data scenario. It is, however, highly simplistic, making the required knowledge to create the relationships accessible to those with limited logical assertion knowledge.

7.4.2.2 Pre-Defined Logical Assertions

An alternative which still enables authors with limited logical assertion knowledge is to provide a meta-model of pre-defined assertions available for selection by the author, such as in Figure 7-8. The author can then build the control loop assertion through model elements just like the document. This approach is more responsive to change as the relationship must now be defined at runtime not design, but the logical assertion is limited to the set of implemented methods.

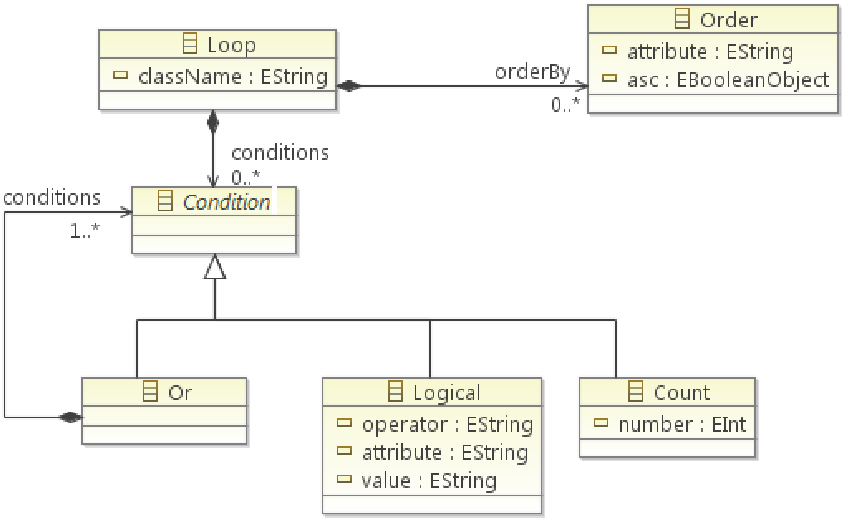


Figure 7-8: Pre-defined Loop Meta-Model

7.4.2.3 Object Query Language

The most flexible approach is to utilise a query language to evaluate the records that should be used. This, however, requires that the author be familiar with the language used and the have the ability to form logical assertions. There are many languages available such as the Object Constraint Language (OCL) or Epsilon Object Language (EOL). At BAE Systems, EOL is favoured. However, as Ecore does not enforce a particular query language, a flexible approach is proposed by having the ability to select the language EOL or OCL and the execution code. Figure 7-9 shows the proposed meta-model for this technique.

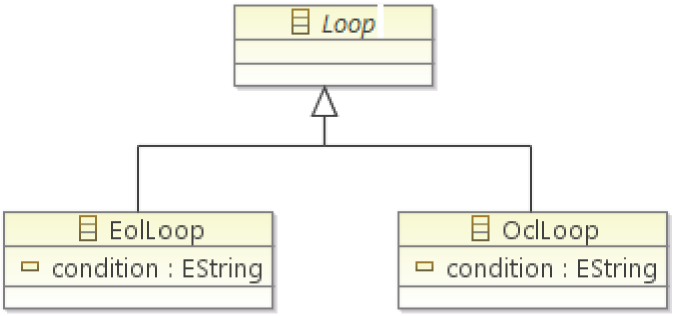


Figure 7-9: Query Language Loop Meta-Model

7.4.3 COMPLETE META-MODEL

Duplicating classes to facilitate templating introduces unnecessary complexity and maintainability issues. The ability to abstract out the semantic reference to a new relationship class provides a low impact update to the meta-model proposed in Section 6.6.6. Given the target domain of complex technical standards and the requirement of an engineer with modelling experience to construct the semantic model, it is considered that the audience would be capable of using the query language based control loop. The abstract does, however, allow the control structure to be extended to facilitate less technical focused methods if required. For example, in Figure 7-8 the Loop class could be implemented as an alternative to EOL/OCL based loops by having the class inherit from the same parent as the EOL/OCL class. Figure 7-10 provides the proposed meta-model to facilitate templating. This meta-model is an excerpt as it does not encompass the whole document model.

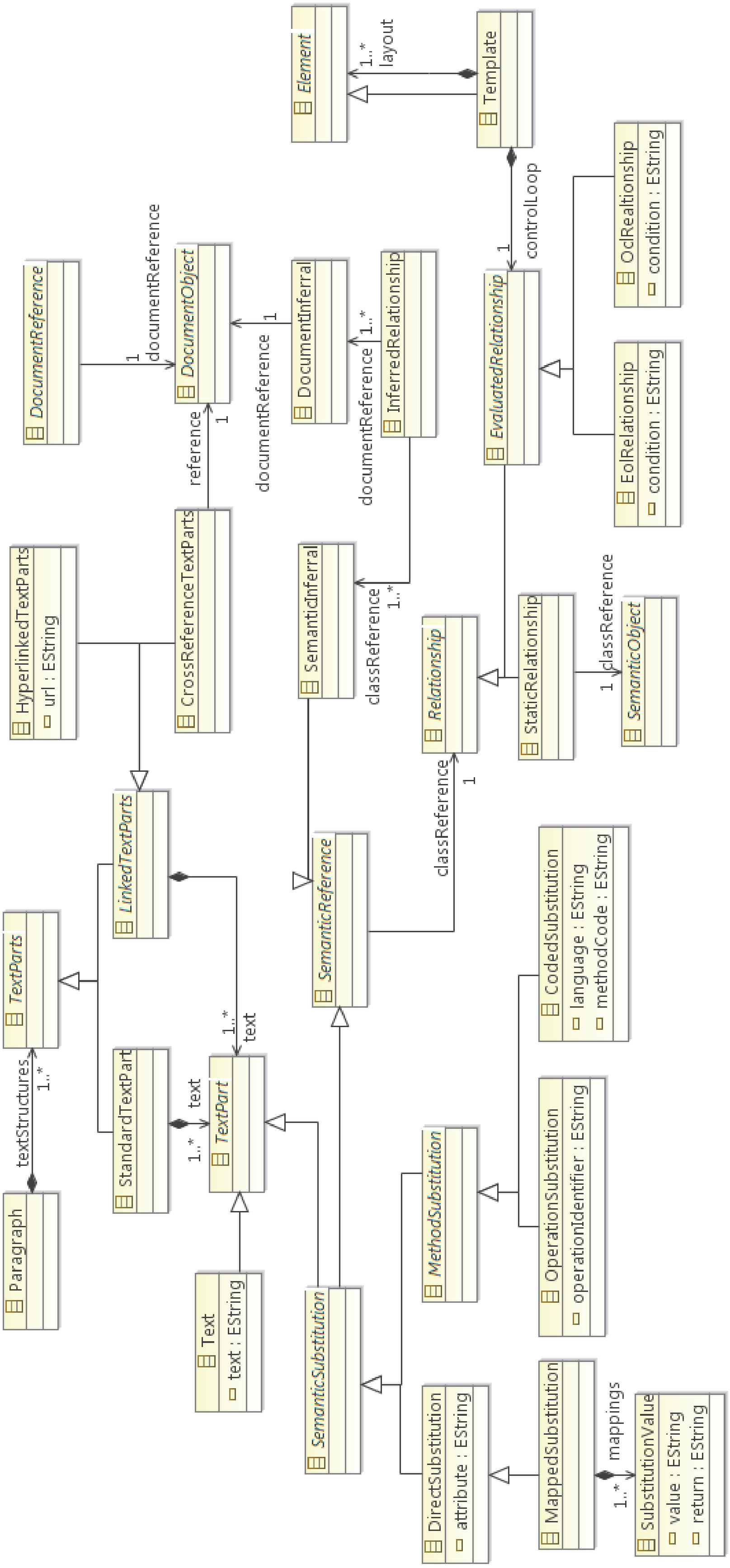


Figure 7-10: Proposed Templating Meta-Model

7.5 EVALUATION

Modern day equipment and infrastructure trends mean that memory and network bandwidth cost and capacities are have much less significance than a decade ago. Given that the PDF size of MIL-STD-6016C is only approximately 15MB, in modern systems this could fit on a mobile phone let alone a computer. Similarly, with network infrastructure moving towards fibre optics, transfer speeds internally can easily achieve 1GB/s, as even broadband networks are being offered with 100MB/s speeds (Ofcom Communications 2013). Therefore costs associated to the storage and communication can readily be ignored.

Although the use of markup does increase the required storage capacity in comparison to a plain text file, what is the effect in relation to alternative document formats? To be usable by an application it must have some form of markup applied. Table 7-2 uses the demonstration paragraph from Figure 7-1 saved as a document using various formats. As can be seen from the table all applications require some form of markup. This comparison is slightly biased as PDF and Microsoft Word do provide advanced features, hence their larger size. It does however, highlight that the approach of hybrid semantic-document models would not be significantly hindered by its markup.

Table 7-2: Document Markup Comparison

File Type	**Bytes**
Plain Text File	449
Model	1,397
PDF	3,606
Microsoft Word	15,847
Microsoft Word XML	76,146

Minification does reduce the required storage capacity. However, the implementation of minification results either in a multi-stage process for both encoding and decoding the semantic model markup or in minifying the document and semantic models. Minifying the model makes understanding and maintaining the models more difficult, as an implied knowledge is required to translate the minified class and attribute names to meaningful names. Both approaches suffer from the loss of readability. Given these factors, minification is not seen as beneficial. Given the analysis of alternative document markup, minification provides no benefit and, therefore, is not recommended.

A drawback of this templating approach is the lack of any clearly defined page concept whilst constructing the document. The amount of information to be displayed is dependent on the evaluation of the control loop. This evaluation is not performed until a rendering request is made, so it is not possible to know the total page count or jump to a particular page within the document model. It is, however, questionable whether the concept of a page should exist. The amount of information to display on a page depends on the dimensions of the page. Therefore the concept of a page should be a flexible concept. Consider a document in a word processor such as Microsoft Word, the total page count and the ability to jump between pages is only evaluated after the definition of the page dimensions. This too can be seen in the use of tablet and e-book readers such as Amazon Kindle, where the page is redefined for the screen size of the device. Therefore, the lack of a clear page object is not seen as being a significant disadvantage.

By making use of a common pattern, the consistency is improved, thus helping achieve benefit B2 (Improve consistency), as the layout and style are applied to each item within the template. Within the MIL-STD-6016C where the repetitively formatted data is large, using templates ensures the data is not rendered differently partway through a common section. Embedding templates within templates improves the consistent rendering further. When the material has these large repetitive sections the amount of effort required to construct the document can be reduced for authors. A large section within the MIL-STD-6016C, such as the data dictionary, can be over 1000 pages. It can be time consuming for an author to enter this much data and it is error prone in ensuring a consistent format and style. By using a template the process can be improved. The format and style only need to be defined once in the template. The control loop then ensures that matching data is not missed. Defining the control loop and template can save a document author considerable amounts of time compared to writing hundreds of independent pages. Time can also be saved during updates and modifications as only the template needs to change and not various points within each repeated format, thus aiding benefit B4 (Easier updating).

Further quality improvements can also be achieved. As with consistency, no data that conforms to the control loop will be forgotten and not rendered, thereby helping to achieve benefit B3 (Identify missing data). This does, however, introduce the new issue of ensuring that the control loop is well thought through and encompasses all intended requirements. This could present issues dependent on the skill and ability for the document author to understand the material and ability to construct a logically defined control loop. Authors may not be familiar with the concept of logical assertions and loops. However, given the target domain of complex technical standards it is assumed that authors have sufficient technical background that this should not be a problem. In addition the semantic meta-model needs to be defined and an author able to construct this should have the necessary skills to construct the loop.

The benefits of this process are not only limited to consistency improvements. In large, technical standards like Link 16, it further helps reduce the amount of required storage space. As discussed earlier (section 7.2), the storage required is for the document model, semantic model and the hybrid links between the two. A template can remove the requirement for hundreds of pages of document and hybrid link markup, further reducing the required storage requirements whilst improving quality and efficiency.

7.6 SUMMARY

The utilisation of templates would reduce the amount of required storage capacity making transfer easier. However, given the continuing advances in reducing the costs associated with storage and network bandwidth capacities, these benefits are considered superficial. The real benefits of templates are achieved by document authors. Templating allows both a consistent look and feel, and the reduction in the risks associated with missed or incorrectly entered data. This results in improved quality control achieving objective A3 (Improve quality control) through benefits B2-4 (Improve consistency, Identify missing data, Easier updating). Efficiencies can also be garnered where the amount of repetitive data is large, as defining the template could be quicker than entering all the data.

8 SUMMARY AND CONCLUSIONS

CHAPTER PREFACE

This chapter concludes the research undertaken by reviewing its achievements against the aims and anticipated benefits. It relates work conducted in earlier chapters to the aims and objectives whilst appraising the ability to fulfil them. Recommendations for future work are presented, before a final summary of the research success.

8.1 CONCLUSIONS

The project had four key objectives. The work included in this thesis has successfully met three out of the four objectives. The level at which these aims could have been met by utilising a single distinct model could be debated, however, a key aspect of this work is the close coupling to the two models to alleviate any synchronisation issues thus aiding the ability to maintain quality by removing any need to maintain multiple copies in different formats that could easier start to differ over time.

Objective A1 had the intention to improve navigation such that users have easier information location, search and recall. The literature in Section 2.5, identified aids that make utilising and navigating electronic documents easier. These aids were added into a web based application in the work described in chapter 4. In terms of usability, the web based application is a significant improvement over existing sources such as the PDF and Microsoft Word documents. The ability to follow hyperlinks and extend the hierarchical menu beyond the top three levels dramatically reduces the amount of clicks engineers are required to make to find certain information. It is accepted that this improvement could have been achieved through modification of the document alone. The creation of the new interfaces to browse the Message Catalogue and Data Dictionary, again greatly reduce the amount of navigation that the user would otherwise have to perform. This also demonstrated Benefit B5 concerning alternative navigation. It is, however, accepted that these interfaces could have been developed with just the semantic model. One of the benefits of the reflective approach described in chapter 5, is that these navigational improvements can be achieved without having to generate alternative sources of the standard which would need to be maintained.

Chapter 6 defined a set of relationships which form the basis of hybrid semantic-document models. In particular Section 6.6.4 defined the inferred relationship. This allows for a set of document model elements to be related to a set of semantic model elements. This can be used to achieve objective A2/Benefit B1 to help the understanding of complex information. The prose within the document model can be linked to its semantic representation. Therefore where the prose appears ambiguous, users can navigate the relationship and view the semantic representation, thus aiding the understanding. Similarly the relationship supports the opposite direction should the prose provide better understanding for the semantic model.

The biggest improvements have been achieved with objective A3 to improve quality control of material. The reflective approach in chapter 5 has meant that the benefits from using a semantic model (Benefit B7 (Model execution facilities)) can be achieved whilst only having to maintain a single source of information. Section 2.7 identified that current model driven web engineering (MDWE) approaches even required a generative step resulting in a synchronisation requirement. This reflective approach means there are no concerns over having to maintain synchronisation between data sources. Although semantically modelling the information alone can aid with benefit B3 in identifying missing data, this approach makes it harder for data to be missed. The addition of Templating (explained in chapter 7) reduces this risk further. By utilising a template and loop controls, it can be guaranteed that items are not missed, provided the loop control is correct. This also has the benefit of improving consistency (Benefit B2) as a template applies a consistent style. Chapter 6 provided the underpinning for Templating, through the definition of the relationships, through using a template the quality can be enhanced further by providing consistent formatting and guaranteeing items are not accidentally omitted. When considered during document creation or modification, a single change to the semantic model is immediately reflected at every location within the document view (provided the relationships have been implemented). This not only further demonstrates Benefit B2 but also highlights Benefit B4 (updating information more effectively).

Objective A4 (and benefit B8 (document comparison)) for enhanced document comparison has not been analysed and is therefore considered unachieved. The SEIC research team started a separate project which utilised the semantic model to demonstrate semantic comparison. It is assumed that as this approach tightly couples a document with a semantic representation, it would be possible to perform semantic level comparisons between documents based on the same semantic meta-model. Therefore, although it is considered unachieved, the approach does indicate that it would allow for enhanced document comparisons. This comparison is particularly useful with the Military Standards domain. As each platform that implements a Tactical Data Link must produce a document that contains all items of the standard that is implemented on the platform, the ability to semantically compare these platform documents could help engineers easily assess the compatibility between platforms. It is also believed that document comparison improvements could be achieved purely through the use of the semantic model. The improvement that this approach brings is the single source. This semantic model resides with the document model. Thus, when using a comparison language such as the Epsilon Comparison Language (ECL), there is no need to maintain a separate source and no concern that the semantic model does not reflect the document when the comparison is made.

8.2 FUTURE WORK

As mentioned in the review of objective A4, document comparison at the semantic level could be hugely beneficial to the Tactical Data Link domain. BAE Systems has already started investigation into this independently. On completion of their research prototype, demonstrations should clearly demonstrate the power of this approach.

There is an area of the TDL Link 16 that has not yet been modelled in a usable manner at BAE Systems. It is referred to by the SEIC as the behavioural modelling aspect. Large sections of the standards define how the system should behave on receipt of certain messages. Analysis has been started in this area by trying to understand what aspects of behaviour are required throughout the document. As this is still early stage research it has not been included in this thesis. However, once a behaviour modelling process has been identified, the relationships defined here should be re-evaluated to ensure they can integrate with the chosen behaviour. Additionally, given a behaviour model, new navigation approaches should be evaluated to review if such a model would make navigating the complex behaviour easier and quicker.

Similarly to document comparison, it is believed that great benefit could be achieved through this approach to implementing an ability to automatically produce document subsets (Benefit B7 (Model execution facilities)). As mentioned earlier, platforms are required to produce their implemented subset of the standard. An approach whereby a user can select the semantic elements that they wish to utilise and then have this hybrid relationship model manage the inclusion of all document elements that are needed to fulfil that semantic collection, would enhance efficiency and potentially the quality (through ensuring elements where not missed) of platform material.

8.3 FURTHER EVIDENCE AND JUSTIFICATION

There has already been work conducted within NATO to improve the rigour of its standards generically by applying XML to the standards. In addition to this starting in mid-2011, effort was made to form a NATO exploratory team for TDL and Inter-operability Modelling for Combined and Multi-National Operations. For which one of the aims from the Technical Activity Proposal for the exploratory team was:

1. Models of TDL behaviour will be developed to enable:
    a. TDL specifications to be interpreted unambiguously.
    b. Inter-operability analysis between pairs of platforms (via their TDL specifications expressed in a new formalism) that is confirmed at the behavioural level.

This work started by NATO is also seen as justification for the need of such an approach as this project. This project has certainly fulfilled the philosophy and overall aim and contributes to the NATO aims. Chapter 6, in particular, has defined relationships that allow the two models to work together and thus information from the semantic model can be rendered in a document view without the need for that information to be duplicated in the document model. Therefore, the philosophy and aim to provide a method to facilitate relationships between text-based document views of information and more rigorous computationally accessible representations of its semantic content has been achieved. The relationships defined have also been produced with consideration for their applicability to wider document models such as those discussed in Section 2.3, which demonstrates the ability to utilise these with mature document models.

The meta-models have purposefully been kept generalised such that this framework can be utilised outside of the case study in other domains. Any domain that has documentation which contains structured semantics could utilise this approach. It is believed, that domains such as nuclear engineering, civil aviation and the automotive industry would benefit from this approach as they too are likely to require computational process and standards that are captured in document form. To a lesser extent material such as the Royal & Ancient’s rules of golf could even utilise this approach. The rules are document based and contain much semantics. This hybrid approach could identify easily where any rule changes conflict or result in missing dependencies.

8.4 THE VALUE OF THE RESEARCH

This research has demonstrated that a document model and a semantic model can be combined to reduce data duplication and thus remove the possibility issues associated with attempting to maintain multiple copies of the documents in multiple formats. This combination enables the prospect of wider benefits, such as to enable documents to be compared at a semantic level rather than textual, and the automatic production of document subsets based on semantics. These improvements and navigational interfaces have enabled BAE Systems to make the MIL-STD-6016C more accessible to their engineers and has resulted in additional research to address the exploitation of the wider benefits.

9 REFERENCES

Acerbis, R., Bongio, A., Brambilla, M. & Butti, S. (2007), WebRatio 5: An Eclipse-based CASE tool for engineering web applications, *in* ‘Proceedings of the International Conference on Web Engineering (ICWE 2007)’, Vol. 4607/2007, pp. 501–505.

Adobe (2008), *Adobe® Framemaker® 9 - MIF Reference*, Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA.

Ahmed, I. & Blustein, J. (2005), Navigation in information space: How does spatial ability play a part?, *in* ‘IADIS International Conference on Web Based Communities’, pp. 119–125.

Alavi, M. & Carlson, P. (1992), ‘A review of MIS research and disciplinary development’, *Journal of Management Information Systems* **8**(4), 45–62.

Ambler, S. (2003), *Agile Database Techniques - Effective Strategies for the Agile Software Developer*, Wiley.

Antoniol, G., Canfora, G., Casazza, G., De Lucia, A. & Merlo, E. (2002), ‘Recovering traceability links between code and documentation’, *IEEE Transactions on Software Engineering* **28**(10), 970 – 983.

Baskerville, R. L. & Wood-Harper, A. (1996), ‘A critical perspective on action research as a method for Information Systems research’, *Journal of Information Technology* **11**(3), 235–246.

Bassey, M. (1999), *Case Study Research in Educational Settings*, Open University Press.

Benbasat, I., Goldstein, D. K. & Mead, M. (1987), ‘The case research strategy in studies of Information Systems’, *MIS Quarterly* **11**(3), 369–386.

Benyon, D., Turner, P. & Turner., S. (2005), *Designing interactive systems*, Addison-Wesley.

Benz, C. (2010), ‘What is DITA and why should you care?’, *Learning Solutions Magazine* .

Blaikie, N. (2009), *Designing Social Research*, 2 edn, Polity Press.

Brugger, R., Zramdini, A. & Ingold, R. (1996), Modeling documents for structure recognition using generalized n-grams, *in* ‘Proceedings of ICDAR’, pp. 56–60.

Burrell, G. & Morgan, G. (1979), *Sociological Paradigms and Organizational Analysis: Elements of the Sociology of Corporate Life.*, Heinemann Educational Books, London.

cable.co.uk (2009), ‘Broadband history’. This is an electronic document. Date retrieved: March 01, 2009. http://www.cable.co.uk/guides/broadband-history/

Calcaterra, A., Antonietti, A. & Underwood, J. (2005), ‘Cognitive style, hypermedia navigation and learning’, *Computers & Education* **44**, 441–457.

Care Technologies Ltd (2010), ‘OlivaNova’. This is an electronic document. Date retrieved: Jan 9, 2010. http://www.care-t.com/products/index.asp

Cáceres, P., Marcos, E., Vela, B. & Juan, R. (2004), A MDA-based approach for web information system, *in* ‘Proceedings of the Workshop in Software Model Engineering Development’.

Ceri, S., Fraternali, P. & Bongio, A. (2000), ‘Web Modeling Language (WebML): a modeling language for designing web sites’, *Computer Networks* **33**(1-6), 137 – 157.

Chen, S. (2002), ‘A cognitive model for non-linear learning in hypermedia programmes’, *British Journal of Educational Technology* **33**(2), 449–460.

Chen, S. Y. & Macredie, R. D. (2002), ‘Cognitive styles and hypermedia navigation: Development of a learning model’, *Journal of the American Society for Information Science Technology* **53**(1), 3–15.

Chopra, V., Li, S. & Genender, J. (2007), *Professional Apache Tomcat 6*, Wrox Publishing Inc, chapter Performance Tuning, pp. 561–584.

Choudrie, J. & Dwivedi, Y. K. (2005), ‘Investigating the research approaches for examining technology adoption issues’, *Journal of Research Practice* **1**(1), Article–D1.

Clowes, D., Dawson, R., Probets, S. & Johnson, J. (2010), Adding quality to web development, *in* R. Dawson, M. Ross & G. Staples, eds, ‘Quality Issues for Today’, The British Computing Society, pp. 123–134.

Clowes, D., Dawson, R., Probets, S., Johnson, J. & Holmes, C. (2008), Pilot studies in using the semantic knowledge of information in large technical documents to aid user navigation, *in* D. Harorimana & D. Watkins, eds, ‘The 9th European Conference on Knowledge Management: ECKM 2008’, Academic publishing, pp. 129–138.

Clowes, D., Dawson, R., Probets, S., Johnson, J., Holmes, C. & Ajit, S. (2009), A multi-layered approach to the development of custom Tactical Data Link applications, *in* ‘Proceedings of the Conference on Systems Engineering Research 2009’.

Cockburn, A., Gutwin, C. & Alexander, J. (2006), Faster document navigation with space-filling thumbnails, *in* 'CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems', ACM, New York, NY, USA, pp. 1–10.

Crowther, D. & Lancaster, G. (2009), *Research Methods*, Butterworth-Heinemann Elsevier.

Dellaert, B. & Kahn, B. (1999), 'How tolerable is delay? consumers' evaluations of Internet web sites after waiting', *Journal of Interactive Marketing* **13**, 41–54.

Dillon, A. (1992), 'Reading from paper versus screens: A critical review of the empirical literature', *Ergonomics* **35**, 1297 – 1326.

Dolin, R. H. & Alschuler, L. (2011), 'Approaching semantic interoperability in Health Level Seven', *Journal of the American Medical Informatics Association* **18**(1), 99–103.

Dolin, R. H., Alschuler, L., Beebe, C., Biron, P. V., Boyer, S. L., Essin, D., Kimber, E., Lincoln, T. & Mattison, J. E. (2001), 'The HL7 clinical document architecture', *Journal of the American Medical Informatics Association* **8**(6), 552–569.

Dolin, R. H., Alschuler, L., Behlen, F., Biron, P. V., Boyer, S., Essin, D., Harding, L., Lincoln, T., Mattison, J. E., Rishel, W. et al. (1999), HL7 document patient record architecture: an XML document architecture based on a shared information model, *in* 'Proceedings of the AMIA Symposium', American Medical Informatics Association, p. 52.

Dolin, R. H., Alschuler, L., Boyer, S., Beebe, C., Behlen, F. M., Biron, P. V. & Shvo, A. S. (2006), 'HL7 clinical document architecture, release 2', *Journal of the American Medical Informatics Association* **13**(1), 30–39.

Dufresne, A. & Turcotte, S. (1997), *Artificial Intelligence in Education: Knowledge and Media in Learning Systems*, Vol. 39, IOS Press, chapter Cognitive Style and its Implications for Navigation Strategies, pp. 287–293.

Dunn, J. (1997), 'Introduction and history of modems'. This is an electronic document. Date retrieved: March 01, 2009. http://www.dementia.org/ julied/tele2100/intro.html

Elm, W. & Woods, D. (1985), Getting lost: A case study in interface design, *in* 'Human Factors Society 29th Annual Meeting', Human Factors Society, pp. 927–931.

Epsilon Eclipse Modeling GMT component (2009), ‘Using EGL as a server-side scripting language in Tomcat’. This is an electronic document. Date retrieved: November 21, 2009. http://www.eclipse.org/gmt/epsilon/doc/articles/egl-server-side/

Fons, J., Valderas, P., Ruiz, M., Rojas, G. & Pastor, O. (2003), OOWS: A method to develop web applications from web-oriented conceptual models, *in* ‘Proceedings of the International Workshop on Web Oriented Software Technology (IWWOST)’, pp. 65–70.

Galliers, R. (1991), *Information Systems Research: Contemporary Approaches & Emergent Traditions*, Elsevier, chapter Choosing Appropriate Information Systems Research Approaches: A Revised Taxonomy, pp. 327 – 345.

Galliers, R. (1992), *Information Systems Research: Issues, Methods and Practical Guidelines*, Blackwell Scientific, chapter Choosing Information Systems research approaches, p. 144.

Glass, R., Ramesh, V. & Vessey., I. (2004), ‘An analysis of research in computing disciplines’, *Communications of the ACM* **47**(6), 89–94.

Gómez, J. & Cachero, C. (2002), *Information modeling for Internet applications*, Idea Group Inc (IGI), chapter OO-H Method: extending UML to model web interfaces, pp. 144–173.

Goede, R. & de Villiers, C. (2003), The applicability of grounded theory as research methodology in studies on the use of methodologies in IS practices, *in* ‘SAICSIT ’03’, pp. 208–217.

Guba, E. (1990), *The Paradigm Dialog*, Sage Publications.

Guba, E. & Lincoln, Y. (1994), *Handbook of Qualitative Research*, Sage, chapter Competing paradigms in qualitative research, pp. 105–117.

Guimbretière, F. (2003), Paper augmented digital documents, *in* ‘Proceedings of ACM symposium on User interface software and technology’, ACM, pp. 51 – 60.

Gulbrandsen, P., Schroeder, T., Mileradand, J. & Nylenna, M. (2002), ‘Paper or screen, mother tongue or English: Which is better’, *Journal of the American Medical Association* **287**(21), 2851–2853.

Guo, J., Takada, A., Tanaka, K., Sato, J., Suzuki, M., Suzuki, T., Nakashima, Y., Araki, K. & Yoshihara, H. (2004), ‘The development of MML (Medical Markup Language) version 3.0 as a medical document exchange format for HL7 messages’, *Journal of Medical Systems* **28**, 523–533.

Gwizdka, J. & Spence, I. (2005), Indirect assessment of web navigation success, *in* 'CHI '05: CHI '05 extended abstracts on Human factors in computing systems', ACM, New York, NY, USA, pp. 1427–1430.

Hasman, A. et al. (2006), HL7 RIM: an incoherent standard, *in* 'Ubiquity: Technologies for Better Health in Aging Societies: Proceedings of MIE2006', Vol. 124, Ios PressInc, p. 133.

Hennum, E. (2005), Specializing domains in DITA, Technical report, IBM.

Hirschheim, R. (1985), 'Information Systems epistemology: An historical perspective', *Research Methods in Information Systems* pp. 13–35.

Holmes, C. & Johnson, J. (2005), Tactical Data Link – domain dictionary, Technical Report SEIC RP 0393, BAE Systems.

Holmes, C., Johnson, J. & Riaz, A. (2007), Tactical Data Links – a description of the supporting models, Technical Report SEIC-RP-0531, BAE Systems.

Holz, H. J., Applin, A., Haberman, B., Joyce, D., Purchase, H. & Reed, C. (2006), 'Research methods in computing: what are they, and how should we teach them?', *SIGCSE Bulletin* **38**(4), 96–114.

Hürsch, W. L. & Lopes, C. V. (1995), Separation of concerns, Technical report, College of Computer Science, Northeastern University.

Hutchinson, J., Whittle, J., Rouncefield, M. & Kristoffersen, S. (2011), Empirical assessment of MDE in industry, *in* 'Proceedings of the 33rd International Conference on Software Engineering', ACM, pp. 471–480.

IBM Corporation (2012), 'IBM Rational DOORS'. This is an electronic document. Date of publication: Jul, 2012. Date retrieved: Jan 11, 2013. http://public.dhe.ibm.com/common/ssi/ecm/en/raf14141usen/RAF14141USEN.PDF

Iivari, J. (1991), 'A paradigmatic analysis of comtemporary schools of IS development', *European Journal of Information Systems* **1**(4), 249–272.

Internet Systems Consortium (2009), 'The ISC domain survey'. This is an electronic document. Date retrieved: March 01, 2009. https://www.isc.org/solutions/survey

Internetworldstats (2008), 'Internet usage world stats'. This is an electronic document. Date retrieved: February 4, 2008. http://www.internetworldstats.com/

Jain, A. & Yu, B. (1998), 'Document representation and its application to page decomposition', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(3), 294–308.

Joan, L., Sundaram, A. & Arumugam, V. (2011), Innovative Evaluation System - IESM: An architecture for the database management system for mobile application, *in* 'ICOMP'11: The 2011 International Conference on Internet Computing'.

Johnson, J. (2007), 'EngD research proposal - hybrid semantic-document models'.

Johnson, J., Holmes, C. & Raiz, A. (2006), 'product standards - tactical data links 2006 final report, Technical Report SEIC-RP-0551, System Engineering Innovation Centre (BAE Systems).

Jones, S., Jones, M., Marsden, G., Patel, D. & Cockburn, A. (2005), 'An evaluation of integrated zooming and scrolling on small screens', *International Journal Human-Computer Studies* **63**(3), 271–303.

Kalantari, R. & Bryant, C. H. (2012), Comparing the performance of Object and Object Relational database systems on objects of varying complexity, *in* L. MacKinnon, ed., 'Data Security and Security Data', Vol. 6121 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 72–83.

Kaplan, B. & Duchon, D. (1988), 'Combining qualitative and quantitative methods in Information Systems research: A case study', *MIS Quarterly* **12**(4), 571–586.

King, A. (2001), 'Extreme HTML optimization'. This is an electronic document. Date of publication: Janurary 10, 2000. Date retrieved: June 9, 2009. Date last modified: March 19, 2001. http://www.webreference.com/authoring/languages/html/optimize/

King, A. (2008), 'Average web page size triples since 2003'. This is an electronic document. Date of publication: April 28, 2008. Date retrieved: June 9, 2009. Date last modified: May 31, 2009. http://www.websiteoptimization.com/speed/tweak/average-web-page/

Klink, S. & Kieninger, T. (2001), 'Rule-based document structure understanding with a fuzzy combination of layout and textual features', *International Journal of Document Analysis and Recognition* **4**, 18–26.

Koch, N. (2000), Software Engineering for Adaptive Hypermedia Systems: Reference Model, Modeling Techniques and Development Process, PhD thesis, LMU Munich.

Koch, N., Meliá-Beigbeder, S., Moreno-Vergara, N., Pelechano-Ferragud, V., Sánchez-Figueroa, F. & Vara-Mesa, J. (2008), 'Model-driven web engineering', *Upgrade* **9**(2), 40–45.

Kolovos, D. S., Paige, R. F. & Polack, F. A. (2006), The Epsilon object language, *in* 'Proceedings of the European Conference in Model Driven Architecture (EC-MDA) 2006', Vol. 4066 of *LNCS*, Bilbao, Spain, pp. 128–142.

Kroiss, C., Koch, N. & Knapp, A. (2009), UWE4JSF: A model-driven generation approach for web applications, *in* 'ICWE '9: Proceedings of the 9th International Conference on Web Engineering', Springer-Verlag, Berlin, Heidelberg, pp. 493–496.

Lister, R. (2005), 'Mixed methods: positivists are from mars, constructivists are from venus', *SIGCSE Bulletin* **37**(4), 18–19.

Lockheed Martin (2012), 'MANDRIL: Message analysis and data reduction for the integration of links'. This is an electronic document. Date of publication: 2012. Date retrieved: Jan 11, 2013. https://secure.lm-isgs.co.uk/MANDRIL/files/Mandril%20Brochure%20210512%20low.pdf

Marchionini, G. (1988), 'Hypermedia and learning: Freedom and chaos', *Educational Technology* **28**, 8–12.

Marcus, A. & Maletic, J. (2003), Recovering documentation-to-source-code traceability links using latent semantic indexing, *in* 'Proceedings of 25th International Conference on Software Engineering, 2003', pp. 125 – 135.

McDonald, S. & Stevenson, R. J. (1996), 'Disorientation in hypertext: The effects of three text structures on navigation performance', *Applied Ergonomics* **27**(1), 61–68.

McDonald, S. & Stevenson, R. J. (1998), 'Navigation in hyperspace: An evaluation of the effects of navigational tools and subject matter expertise on browsing and information retrieval in hypertext', *Interacting with Computers* **10**(2), 129–142.

McDonald, S. & Stevenson, R. J. (1999), 'Effects of text structure and prior knowledge of the learner on navigation in hypertext', *Human Factors* **40**, 18–27.

McKnight, C., Dillon, A. & Richardson, J. (1990), *Hypertext: state of the art*, Intellect, Oxford, chapter A Comparison of linear and hypertext formats in information retrieval, pp. 10–19.

Meliá, S. & Gómez, J. (2006), 'The WebSA approach: Applying model driven engineering to web applications', *Journal of Web Engineering* **5**(2), 121–149.

Mingers, J. (2001), 'Combining IS research methods: Towards a pluralist methodology', *Information Systems Research* **12**(3), 240–259.

Mingers, J. (2003), 'The paucity of multi-method research: a review of the Information Systems literature', *Information Systems Journal* **13**, 233–249.

Miniwatts Marketing Group (2009), 'Internet growth statistics'. This is an electronic document. Date retrieved: March 01, 2009. http://www.internetworldstats.com/emarketing.htm

Mitchell, T., Chen, S. & Macredie, R. (2004), Adapting hypermedia to cognitive styles: Is it necessary?, *in* 'Proceedings of Workshop on Individual Differences in Adaptive Hypermedia', Published online by Department of Computer Science and Information Systems Birkbeck University of London, pp. 70–83.

Moreno, N. & Vallecillo, A. (2008), 'Towards interoperable web engineering methods', *Journal of the American Society for Information Science and Technology* **59**(7), 1073 – 1092.

Morrison, J. & George., J. (1995), 'Exploring the software engineering component in MIS research', *Communications of the ACM* **38**(7), 80–91.

Muter, P., Latremouille, S., Treurniet, W. & Beam, P. (1982), 'Extended reading of continuous text on television screens', *Human Factors* **24**(5), 501–508.

Muter, P. & Maurutto, P. (1991), 'Reading and skimming from computer screens and books: The paperless office revisited?', *Behaviour & Information Technology* **10**(4), 257–266.

Nah, F. (2004), 'A study on tolerable waiting time: how long are web users willing to wait?', *Behaviour & Information Technology* **23**(3), 153–163.

NATO (2001), Technical characteristics of the multifunctional information distribution systems (MIDS), Technical report, NATO.

NATO (2006), Tactical data exchange – link 16, Technical Report STANAG No. 5516, NATO Standardization Agency.

Nielsen, J. (1990), 'The art of navigating through hypertext', *Communications of the ACM* **33**(3), 296–310.

Nielsen, J. (1996), ‘Original top ten mistakes in web design’. This is an electronic document. Date retrieved: June 9, 2009. http://www.useit.com/alertbox/9605a.html

Nielsen, J. (2004), ‘The need for speed’. This is an electronic document. Date retrieved: June 9, 2009. http://www.useit.com/alertbox/9703a.html

Nielsen, J. (2007), ‘Top ten mistakes in web design’. This is an electronic document. Date retrieved: March 01, 2009. http://www.useit.com/alertbox/9605.html

OASIS (2009), ‘The DocBook schema version 5.0’.

OASIS (2010), ‘Darwin Information Typing Architecture (DITA) version 1.2’.

Object Management Group (2006), ‘Object Constraint Language, version 2.0’. This is an electronic document. Date retrieved: November 21, 2009. http://www.omg.org/spec/OCL/2.0/PDF

Ofcom Communications (2013), ‘Average UK broadband speed continues to rise’. This is an electronic document. Date retrieved: November 1, 2013. http://media.ofcom.org.uk/2013/08/07/average-uk-broadband-speed-continues-to-rise/

Ogheneovo, E. E., Asagba, P. O. & Ogini, N. O. (2013), ‘An object relational mapping technique for Java framework’, *International Journal of Engineering Science Invention* **2**, 01–09.

O’Hara, K. & Sellen, A. (1997), A comparison of reading paper and on-line documents, *in* ‘CHI ’97: Proceedings of the SIGCHI conference on Human factors in computing systems’, ACM, New York, NY, USA, pp. 335–342.

Oracle (1993), ‘Javadoc Technology’. This is an electronic document. Date retrieved: November 1, 2013. http://docs.oracle.com/javase/7/docs/technotes/guides/javadoc/

Orlikowski, W. & Baroudi, J. (1991), ‘Studying Information Technology in organizations: Research approaches and assumptions’, *Information Systems Research* **2**, 1–28.

Palmer, J. (2002), ‘Designing for web site usability’, *IEEE Computer* **35**, 102–103.

Palmirani, M., Contissa, G. & Rubino, R. (2009), Fill the gap in the legal knowledge modelling, *in* G. Governatori, J. Hall & A. Paschke, eds, ‘Rule Interchange and Applications’, Vol. 5858 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 305–314.

Park, J. & Kim, J. (2000), Effects of contextual navigation aids on browsing diverse web systems, *in* 'CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems', ACM, New York, NY, USA, pp. 257–264.

Piolat, A., Roussey, J. & Thunin, O. (1997), 'Effects of screen presentation on text reading and revising', *International Journal on Human-Computer Studies* **47**, 565–589.

Price, M. N., Golovchinsky, G. & Schilit, B. N. (1998), Linking by inking: Trailblazing in a paper-like hypertext, *in* 'Proceedings of UK Conference on Hypertext', ACM, pp. 30–39.

Rahtz, S. & Cummings, J. (2013), 'Getting started using TEI: What is TEI?'. This is an electronic document. Date retrieved: November 01, 2013.

Ramesh, B. & Jarke, M. (2001), 'Toward reference models for requirements traceability', *IEEE Transactions on Software Engineering* **27**(1), 58 –93.

Reason, P. & Bradbury, H. (2001), *The Handbook of Action Research: Participative Inquiry and Practice Second Edition*, Sage Publications.

reStructuredText (2010), 'reStructuredText - markup syntax and parser component of docutils'. This is an electronic document. Date retrieved: November 1, 2013. http://docutils.sourceforge.net/rst.html

Robson, C. (2002), *Real world research: A resource for social scientists and practitioner-researchers*, Blackwell.

Rose, G., Meuter, M. & Curran, J. (2005), 'On-line waiting: The role of download time and other important predictors on attitude towards e-retailers', *Psychology & Marketing* **22**(2), 127–151.

Rose, L. M., Paige, R. F., Kolovos, D. S. & Polack, F. A. (2008), The Epsilon Generation Language (EGL), *in* 'Proceeding of the European Conference in Model Driven Architecture (ECMDA)'.

Rouet, J.-F. & Bigot, L. L. (2007), 'Effects of academic training on metatextual knowledge and hypertext navigation', *Metacognition and Learning* **2**, 157–168.

Salminen, A., Kauppinen, K. & Lehtovaara, M. (1997), 'Towards a methodology for document analysis', *Journal of the American Society for Information Science* **48**(7), 644–655.

Saunders, M., Lewis, P. & Thornhill, A. (2009), *Research methods for business students*, 5 edn, Pearson Education Limited.

Schadow, G. & Mead, C. N. (2006), The HL7 reference information model under scrutiny, *in* ‘Ubiquity: technologies for better health in aging societies: proceedings of MIE2006’, Vol. 124, Ios PressInc, p. 151.

Schilit, B. N., Golovchinsky, G. & Price, M. N. (1998), Beyond paper: Supporting active reading with free form digital ink annotations, *in* ‘CHI ’98: Proceedings of the SIGCHI conference on Human factors in computing systems’, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, pp. 249–256.

Schumacher, G. & Waller, R. (1985), *Designing Usable Texts*, Academic Press, Inc., chapter Testing Design Alternatives: A Comparison of Procedures, pp. 377–403.

Scotland, J. (2012), ‘Exploring the philosophical underpinnings of research: Relating ontology and epistemology to the methodology and methods of the scientific, interpretive, and critical research paradigms’, *English Language Teaching* **5**(9), 9–17.

Selvidge, P., Chaparro, B. & Bender, G. (2002), ‘The world wide wait: Effects of delays on user performance’, *International Journal of Industrial Ergonomics* **29**, 15–20.

Shubho, A.-F. (2009*a*), ‘Top 10 steps to optimize data access in SQL Server: Part i (use indexing)’. This is an electronic document. Date of publication: April 28, 2009. Date retrieved: July 9, 2009. http://www.codeproject.com/Articles/34372/Top-10-steps-to-optimize-data-access-in-SQL-Server

Shubho, A.-F. (2009*b*), ‘Top 10 steps to optimize data access in SQL Server: Part ii (re-factor TSQL and apply best practices)’. This is an electronic document. Date of publication: June 1, 2009. Date retrieved: July 9, 2009. http://www.codeproject.com/Articles/35665/Top-10-steps-to-optimize-data-access-in-SQL-Server

Souders, S. (2008), ‘High-performance web sites’, *Communications of the ACM* **51**(12), 36–41.

Spanoudakis, G., Zisman, A., Pérez-Miñana, E. & Krause, P. (2004), ‘Rule-based generation of requirements traceability relations’, *Journal of Systems and Software* **72**(2), 105 – 127.

Stonebraker, M. & Moore, D. (1995), *Object Relational DBMSs: The Next Great Wave*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Sun, L. & Guimbretière, F. (2005), Flipper: A new method of digital document navigation, *in* 'CHI '05 extended abstracts on Human factors in computing systems', ACM, New York, NY, USA, pp. 2001–2004.

Sun Microsystems (2009), 'JSR-000315 Java Servlet 3.0 specification'. This is an electronic document. Date retrieved: November 21, 2009. http://jcp.org/aboutJava/communityprocess/pr/jsr315/index.html

Taal, M. & Irawan, H. (2012), 'Teneo'. This is an electronic document. Date retrieved: November 21, 2013. http://wiki.eclipse.org/Teneo

TEI Consortium (2013), *TEI P5: Guidelines for Electronic Text Encoding and Interchange*, 2.50 edn, TEI Consortium, 30 Addison St. Arlington, MA 02476 USA.

The Eclipse Foundation (2009*a*), 'About the Eclipse Foundation'. This is an electronic document. Date retrieved: November 16, 2009. http://www.eclipse.org/org/

The Eclipse Foundation (2009*b*), 'Eclipse Modeling Framework Project (EMF)'. This is an electronic document. Date retrieved: November 16, 2009. http://www.eclipse.org/modeling/emf/

The Eclipse Foundation (2009*c*), 'Package org.eclipse.emf.ecore'. This is an electronic document. Date retrieved: November 16, 2009. http://download.eclipse.org/modeling/emf/emf/javadoc/2.9.0/org/eclipse/emf/ecore/package-summary.html#details

U.S. Department of Defense (2004), Department of Defense interface standard – Tactical Data Link (TDL) 16 – message standard, Technical Report MIL-STD-6016C, US Department of Defense.

Vallecillo, A. et al. (2007), MDWEnet: A practical approach to achieving interoperability of Model-Driven Web Engineering methods, *in* 'Workshop Proceedings of 7th International Conference on Web Engineering', pp. 246 – 254.

Vanathi, B. & Uthariaraj, V. R. (2011), 'Comparing and choosing appropriate database for storing context in context aware system', *Journal of Computer Science* **7**, 1027–1032.

Vaughan, M. W. & Dillon, A. (2006), 'Why structure and genre matter for users of digital information: A longitudinal experiment with readers of a web-based newspaper', *International Journal of Human-Computer Studies* **64**(6), 502–526.

Ventura, C. (1988), Why switch from paper to electronic manuals?, *in* ‘Proceedings of ACM Conference on Document Processing Systems 1988’, ACM, pp. 111–116.

VisualWade (2006), ‘VisualWade’. This is an electronic document. Date of publication: April 28, 2008. Date retrieved: Jan 9, 2010. Date last modified: November 2006. http://www.visualwade.com/home.php

Vreede, G. d. (1995), Facilitating Organisational Change: The Participative Application of Dynamic Modelling, PhD thesis, Delft University of Technology.

Walsh, N. (2010), *DocBook 5: The Definitive Guide*, O’Reilly Media.

Weinreich, H., Obendorf, H., Herder, E. & Mayer, M. (2006), Off the beaten tracks: Exploring three aspects of web navigation, *in* ‘Proceedings of the 15th international conference on World Wide Web 2006’, ACM, New York, NY, USA, pp. 133–142.

Wellner, P. (1993), ‘Interacting with paper on the DigitalDesk’, *Communications of the ACM* **36**(7), 86–97.

Willis, J. W. (2007), *Foundations of Qualitative Research Interpretive and Critical Approaches*, SAGE Publications, Inc, chapter World Views, Paradigms, and the Practice of Social Science Research.

Winter, S. & Nittel, S. (2003), ‘Formal information modelling for standardisation in the spatial domain’, *International Journal of Geographical Information Science* **17**(8), 721–741.

Wolfram (2013), ‘What is computable document format™(CDF)?’. This is an electronic document. Date retrieved: November 1, 2013. http://www.wolfram.com/cdf/

Yahoo Inc. (2009), ‘Best practices for speeding up your web site’. This is an electronic document. Date retrieved: March 01, 2009. http://developer.yahoo.com/performance/rules.html

Zeigler, B. (2006), ‘Simulation-based testing of emerging defense Information Systems’, Presentation [Available at: http://www.acims.arizona.edu/PUBLICATIONS/Presentations/AuburnTalk.ppt].

Appendix A COMPUTER STANDARDS & INTERFACES JOURNAL ARTICLE

Paper published in 2013 in Journal Computer Standards & Interfaces.36 (1), pp.97-109.

Extending Document Models to Incorporate Semantic Information for Complex Standards

Darren Clowes[a], Ray Dawson[a], Steve Probets[a]

[a]Department of Computer Science, Loughborough University, UK.

ABSTRACT

This paper presents the concept of hybrid semantic-document models to aid information management when using standards for complex technical domains such as military data communication. These standards are traditionally text based documents for human interpretation, but prose sections can often be ambiguous and can lead to discrepancies and subsequent implementation problems. Many organisations will produce semantic representations of the material to ensure common understanding and to exploit computer aided development. In developing these semantic representations, no relationship is maintained to the original prose. Maintaining relationships between the original prose and the semantic model has key benefits, including assessing conformance at a semantic level rather than prose, and enabling original content authors to explicitly define their intentions, thus reducing ambiguity and facilitating computer aided functionality. A framework of relationships is proposed which can integrate with common document modeling techniques and provide the necessary functionality to allow semantic content to be mapped into document views. These relationships are then generalised for applicability to a wider context.

Keywords: *Semantic Modeling; Information Modeling; Aerospace; Defense; Quality*

1 INTRODUCTION

Increasingly the design of complex engineered products and systems (here within referred to as systems) are becoming more reliant on computer-supported models/representations of information, which can be used with computer checking algorithms to ensure consistency and correctness. Computer-supported models enable engineers to represent information graphically, which can aid understanding. These models also give a more rigorous definition of the systems requirements than that of the more ambiguous (open to interpretation) prose representation of the same information. Many system properties can be modeled, such as, behaviour, functionality, verification information, and manufacturing instructions.

For the purpose of this research, a complex engineering domain refers to a domain where the documented materials are large in volume and contain elements that could be modeled semantically. These elements leave no possibility for different interpretations; as such they can be modeled to create the semantic information. Semantic information is the formalised description of the meaning and relationship between elements. This information is often represented in a semantic model. In particular, we use the term semantic model to refer to our models of the case study domain such as the message catalogue and data dictionary. This differs from document model which is a term used to describe a model which represents the structure of the document, i.e. paragraphs, tables etc. This model does not imply any knowledge of the information, more the spatial rendering of the information.

In engineering domains, the documentation is normally highly structured. Engineers are required to use, and have a detailed knowledge of these documents to perform their task. In complex engineering domains, these documents normally consist of large sections of prose. It can, therefore, be hard to understand the precise objective of the section. These sections generally contain elements which could be used semantically mixed with prose which aims to help the readers understanding, but can often be ambiguous.

There are many sectors of the complex engineering domain that could benefit from the use of semantic and static representation of information. The types of domains that will benefit include those that have associated standards documents. The defence industry contains many such documents. Other industries that have similar documents, which this technique may benefit, include the automotive, aerospace, nuclear and legislative sectors.

Most military standards in use are still disseminated as text-based documents. In our experience, this is also the case in most other domains requiring complex detailed standards. These standards can be thousands of pages in volume, which can make locating and composing information in them challenging and laborious. Within the development of military hardware, engineers are typically required to produce text based documents based on a subset of the standards implemented by their product. Product testing will require the engineers to promptly locate information in the standard and validate that their product conforms to it. Interoperability is also crucial in military applications where engineers must verify that their implementation does not impede or conflict with other products. To address these issues, we have found it useful to extract semantic models represented in a structured format which are then amenable to automated querying and processing. In our experience this approach can greatly enhance both the accuracy and speed of locating and composing information from different parts of a standard. We have also found that in order to make models useful for engineers, it is essential to construct a suitable and familiar user interface for querying and navigating them.

2 BACKGROUND

To evaluate the proposal of hybrid semantic-document models, the military domain of Tactical Data Links (TDL) was chosen as a suitable case study with the support of BAE Systems. Tactical Data Links display many of the characteristics of a complex engineering domain. The domain is specified through standard documents, each of which is many thousands of pages in length. They are highly structured and contain material that can be represented semantically. Regulation of conformance is obtained through production of documents similar in format and style to the original standard.

The TDL domain is a collection of related technologies designed to work within the command, control, communications, computers, and intelligence (C4i) used in the dissemination of information within a battlespace to support joint and combined operations. The TDL domain is often referred to as a family of standards, as several variations of Tactical Data Links have evolved to interface with specific unit types (infantry, aircraft, ships, etc). These variations may differ in waveforms, bandwidths, protocols and capabilities.

The TDL provides one of the backbone technologies underpinning the defence community's goal of Network Enabled Capability (NEC) by providing the information and infrastructure to afford users with both an integrated picture of the battlefield and also provide tasking orders and responses. A number of TDLs are in service with coalition forces, and are implemented on a variety of assets, such as aircraft, ships, land vehicles and command stations, an example of which is illustrated in Figure 1.

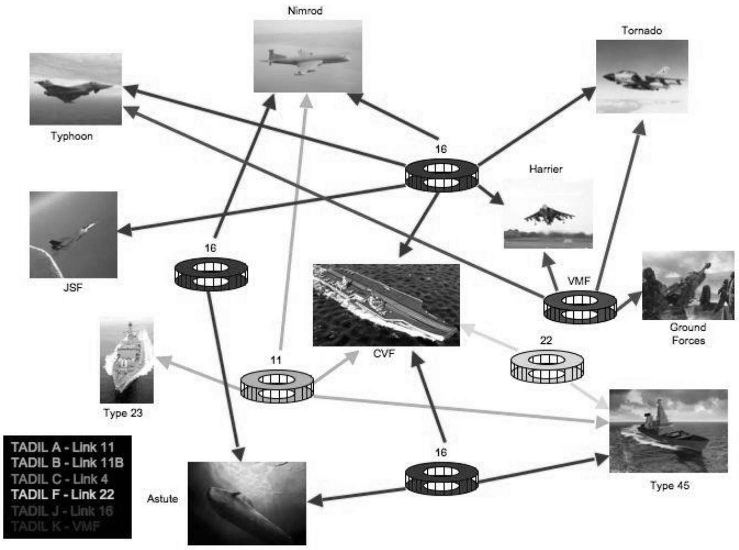


Figure 1. TDL Overview

For the purposes of this project it has been decided to use the Link 16 standard. This is described in the U.S. Department of Defense Military Standard MIL-STD-6016C (U.S. Department of Defense 2004) and NATO Standardization Agreements (STANAGs) 5516 (NATO 2006) and 4175 (NATO 2001). As both STANAGs are required for the complete representation of Link 16, the MIL-STD version will be the basis for this project's interpretation.

The Link 16 TDL is a general purpose TDL, in contrast to some others, e.g. Link 4A or Variable Message Format (VMF). A list of data link characteristics is provided elsewhere (Holmes & Johnson 2005) . Link 16 has evolved over a number of years, stemming from a requirement identified by the US military in the early 1970s for a TDL offering a broad range of functions that would be applicable for use across multiple forces (e.g. Navy, Marines, Air Force, Army, etc.).

The Link 16 TDL (in MIL-STD and STANAG) is described in the form of narrative combined with many tables and relatively few figures. It is known to feature a number of shortcomings affecting its usability (Zeigler 2006). Of particular relevance to the research undertaken by the System Engineering Innovation Centre at Loughborough are, that the standard:

- is document-based, with no apparent underpinning model
- is largely narrative
- is open to (mis)interpretation
- is not checkable by machine
- contains duplication of material, inviting inconsistency
- has only very limited use of hyperlinking, impeding document navigability
- comprises many interdependent sections and appendices
- is enormous, >8000 pages

3 WHAT IS THE PURPOSE?

The aim of hybrid semantic-document models is to provide a method to facilitate relationships between text-based document views of information and more rigorous computationally accessible representations of its semantic content. Regulatory requirements enforce a document-centric culture, and engineers themselves come from many engineering disciplines and generally consider document based standards or requirements as familiar and comfortable. Although military standards are highly structured and contain many elements that could be modeled symantically, they are not in a format accessible to computer processing. Computer aided processing of standards offers several beneficial processes that can enhance the efficiency and productivity of engineers using the standards. However, a solely computer modeled standard becomes inaccessible to the engineers.

Therefore, this philosophy aims to achieve a level, whereby the engineers can generate the respective views that they are comfortable with (i.e. traditional document based), as well as having a sound validated semantic model which is computer readable and executable. In reviewing this approach, there are two perspectives to consider when applying the philosophy. Firstly, managing legacy information maintained in a prose/non-model based document and, secondly, the generation of new documents to aid in the consistency and transfer of understanding and knowledge. It is, therefore, believed that such an approach of hybrid semantic-document models will facilitate the following benefits and objectives:

> **Capture the interpretation of complex prose in a more formal mechanism.** This allows the information author to express their intended action explicitly (removing ambiguity) or where the document already exists, an expert can capture their interpretation which can then be shared by document users to gain a shared consistent knowledge.
>
> **Identify and resolve/prevent inconsistencies within the document.** Mapping semantic data to its required positions within a document reduces the possibility of errors of inconsistency being introduced through duplicated data.
>
> **Identify and resolve/prevent missing data.** It is anticipated that by modeling the information, cases where references to non-existent data can be captured, resolved and/or avoided. The use of model validation tools would enhance this ability to detect these instances.
>
> **Update information more effectively.** An update to the semantic model would be reflected in all positions of the document. This should make updating and maintaining the document easier.
>
> **Provide alternative navigation structures.** It would be possible to provide alternative navigation menus to the traditional tree like hierarchical document structure. Similarly it would be possible to restrict the amount of information displayed by restricting the information according to semantic constraints rather than document hierarchies.

Enhance document information searching. It should be possible to enhance traditional document searching through the use of the semantics. Search methods could involve querying the semantic models to return related points within the document.

Use model execution facilities. By having the core semantic information of the document modeled, it should be possible to utilise model execution facilities to gain computational benefits from the document. This could include code generation, model querying, model validation, etc.

Perform advanced document comparison. It would be possible to compare the semantics of multiple documents to assess their compatibility and/or similarity beyond the simple text comparison available for a traditional document.

This research, therefore, is carried out to help resolve (1) the knowledge problems of understanding complex documents, interpreting their meaning in particular context and identifying errors of inconsistency and completeness, and (2) the design problem of reforming such complex documents to achieve completeness, eliminate inconsistencies and enable computer execution [6]. In respect of (2) the resulting hybrid semantic-document models can be considered to be a design artifact, though this is not their only purpose as the knowledge problems of (1) indicate.

4 WHY IS IT NEEDED?

The primary user base for this research is a wide range of engineers who may not be familiar with modeling and model techniques. However, our audience is highly familiar with their traditional document centric methods of information capture and rendering. They would gain much from utilising formal models but regulation and business practises require much of their utilisation to be through a document. A document alone can be formally specified in terms of its structure, but the information cannot. The document can be ambiguous, creating problems of consistency and completeness when they grow large.

Semantic modeling is a very well-known and board research field, a semantic model alone would provide several of the anticipated benefits such as, model execution, avoiding missing data, etc., but they do not have the traceability back to the originating information source (in our case a document). They also do not facilitate easy rendering of information in document format for regulation and engineers' understanding. Document markup languages provide the easy rendering of information, but do not provide the semantic structure to model the information in a domain specific way. Therefore, in this proposal we try to address the combination of these two areas.

It is possible to maintain separate semantic models and document markup models, however, a second data source is introduced which must be maintained in synchronisation with the original document otherwise the benefits are lost completely. With separate models, it is also difficult to indicate what part of the model relates to elements of prose in the document, this means it would be difficult for engineers with no modeling background to understand.

There appears to be little research in combining a formal model with document markup. Much investigation has been carried out into defining models and/or ontologies to add meaning to complex documents [7, 8], but this does not help with the execution or other model management functions such as code generation or model validation. By using predefined domain markup, more generalised queries can be applied to the document which would only move part way to our overall aim. It is predominantly a technique to assist with information retrieval tasks, for which much has been discussed on how to model the document structure to aid the retrieval process. This too, does not help in achieving our aim.

Similarly much research has been conducted into requirements traceability [9, 10, 11, 12]. Although you could consider the MIL-STD as a requirements document, the research on requirements traceability is focused on tracing between chains of documents. Even if we treated our semantic model as an XML file to facilitate requirements traceability techniques, it still would not help reduce the duplication and would make utilising the models more difficult.

One of closest examples of related work is by Winter and Nittel [13] who discuss the geospatial standard and address the need for more formal modeling. Although they do stress the relationship between prose and the models, there is no direct traceability and the information is duplicated. They make use of functional programing languages which, given the wide range of engineer skills anticipated to use our solution, would be likely to be confusing. During workshops with a cross section of engineers using the MIL-STD, it was found that even UML style modeling was not widely understood.

The closest related area of research is the Health Level 7 Clinical Document Architecture (CDA) framework [14, 15, 16, 17]. The framework has two core parts, the Clinical Document Architecture (CDA) and the Reference Information Model (RIM). This is equivalent to our proposed approach for complex engineering domains, a document model similar to the CDA and a semantic model representing the RIM. The aims of the framework are not too dissimilar to our research:

> “The need for a clinical document standard stems from the desire to unlock the considerable clinical content currently stored in free-text clinical notes and to enable comparison of content from documents created on information systems of widely varying characteristics. Given the variability in clinical notes, including structure, underlying information models, degree of semantic encoding, use of standard healthcare terminologies, and platform- and vendor-specific features, it is currently difficult to store and exchange documents with retention of standardized semantics over both time and distance.” [14]

The RIM provides a vocabulary (ontology) and the meta-model to construct semantic information within a document. This inclusion of semantic information in the document is a primary aim for our research. The CDA is a document markup standard that specifies the structure and semantics of clinical documents. Explicit markup tags are defined that facilitate the encoding of text according to the ontology. The text is not directly retrieved form the underlying RIM, the ontology is used purely to encode the text string. In addition to utilising the ontology, the CDA also allows for the recording of process and interactions through the use of instances of the RIM model. These instances of the RIM require the reuse of the data recorded in the textual elements leading duplication. There is no attempt made to reutilise the information recorded in either part, although CDA does record both the instances of both models within the one document.

There is criticism of the RIM [18, 19], however, the RIM itself provides no benefit to our domain as it is an information model for clinical workflows. The CDA and RIM at present have also deliberately avoided advanced and complex semantics such that a staged approach to the implementation and delivery of a shared structure and semantics can be achieved [14]. This too is something that we consider suitable for complex engineering domains by growing and evolving the semantic model over a period of time allowing for user adoption and providing time to establish the complex semantics.

An astute feature provided with the CDA is the ability to locally define markup [16] which can extend the CDA semantics provided that these extensions do not change the meaning of any of the standard data items, and can be safely ignored by applications without the local knowledge. The CDA approaches the problem where there are multiple sources of information that need to be standardised to enhance interoperability. Our domain works in the opposite direction. The source is singular and well controlled, we however require better methods to disseminate the information to different users with different requirements.

5 CASE STUDY ANALYSIS

Tactical Data Link 16 is one such standard that the authors believe would benefit from the use of hybrid semantic-document models. Within the Military Standards area alone there is a wealth of standards that display similar characteristics to the Link 16 standard, for example VMF, Link 11. In addition NATO and the US DoD produce these standards across a differing set, i.e. Link 16 is defined by US DoD MIL-STD-6016 [1] but NATO define the standard with STANAGs 5516 [2] and 4175 [3]. These standards themselves are huge documents thousands of pages in volume.

Given the wealth of material available and due to the close association between the researchers and the study, the approach was an action research [20] based “single-case” case study [21] method, focusing on the MIL-STD-6016C. Given the volume of the MIL-STD-6016C the authors consider a single-case design sufficient to allow the construct and proof of the hybrid semantic-document models concept. It is also believed that the framework developed by analysing the MIL-STD-6016C can thus be generalised such that it can be applied not just to STANAGs or Military Standards, but to other domains where the capture of complex rigorous information is maintained in a prose based document.

It is intended to produce a framework that will enable the concept of hybrid semantic-document models to be applied to multiple sources. The framework in this context refers to a collection of model components with guidance on where and how they can be used to construct a hybrid semantic-document model of a whole document. Grounded theory [22] is the discovery of a theory through the systematic analysis of data. In this research we have started with the concept of hybrid semantic-document models and, by using a grounded theory-like approach the aim was to discover the relationships and means of representing these relationships from the MIL-STD-6016C case study to produce this framework.

These methods will be applied and evaluated using an interpretivist philosophy. An interpretivist philosophy is being employed given the multitude of potential valid interpretations of the data and formulation of frameworks. Within this, a constructive approach as per Iivari [23] will be used to define the framework. This is intended to utilise an inductive process by observing the material present in the Tactical Data Link Standards and using these observation to infer a framework which can achieve the objectives in Section 3.

In conducting the case study, a significant form of data capture will be required. To obtain this data we shall use a document analysis method similar to that of Salminen et al [24]. They use multiple sources as part of their document analysis, however it is the authors' opinion that the volume of MIL-STD-6016C and its segregation into discrete sections is sufficient for data gathering. Given the volume, sampling was utilised. The sampling method was performed using cluster sampling. Given prior knowledge of the document's format and structure, random samples were picked to highlight the types of document and information likely to be encountered. The sampling method was performed using cluster sampling. Given prior knowledge of the document's format and structure, random samples were picked to highlight the types of document and information likely to be encountered.

5.1 Sample Selection

Previous extraction and modeling of the semantics at BAE Systems has led to a level of knowledge of the document. As such, a selection of pages from the standard was selected using stratified sampling, key areas of the material can be grouped into strata, and a random sample selected from these strata that represent the semantic model and/or the document structure. The strata were not equivalent in volume, but were intended to cover a cross section of the differing document structures and the semantics that would need to be mapped. For security and sensitivity reasons actual headings and text used in this paper have been substituted with fictional material that maintains the essence of the original.

Three pages from each strata were selected to form the sample. Should the selected page contain a full page graphic, this was ignored and an alternative random page selected. Consecutive pages were also prevented. The random sampling method was non-probabilistic judgement. Judgement was provided by a member of the BAE Systems research team who had knowledge of both the semantic models and the document sections and content. The samples obtained are shown in Table 1.

Table 1. Sample Selection

Strata	Pages
Payload definitions	B-38, B-146, B-824
Message definitions	5.1-15, 5.1-440, 5.2-12
Message rules	F-6, K-14, P-53
Transmit and receive tables	5.4-J7.0-6, 5.4-J12.6-1, 5.5-J14.2-1
Constraint information,	4-23, 4-29, 4-169
General information	1-17, 3-56, 4-130
Message Uses	5.5-J12.0-15, 5.5-J0.3-1, 5.5-J2.5-2
Minimum Specification	W-18, W-27, W-92

On analysis of the samples, it became apparent that, for the message rules, too much meaning and information was lost by reading a single page. The sample selection for message rules was, therefore, changed to a trace. A non-probabilistic random rule was used and all pages necessary to trace an understanding were selected. For example, Rule F.1.2 covers two pages, however, through its references it requires further knowledge of two rules, a transmit table and indirect knowledge of four other messages. These indirect references were ignored as their exclusion would not prevent basic understanding. This resulted in a total of six pages for analysis for this rule.

In maintaining two sources of information and formatting, the primary location and relationships between sources needs to be considered. The semantic model is the obvious choice as the primary information location. As the document model contains multiple references to the same object, this information can be maintained singularly in the semantic model, resulting in a change to the semantics being reflected at every location in the document. This supports the objective of reduced duplication and hence improves quality. It is also the model which is likely to experience the most computational activity as the document is simply a rendering activity.

However, is it that simple? The two models can be applied to a matrix to aid the analysis. This two-by-two matrix is represented in Figure 2.

5.2 Location of Information

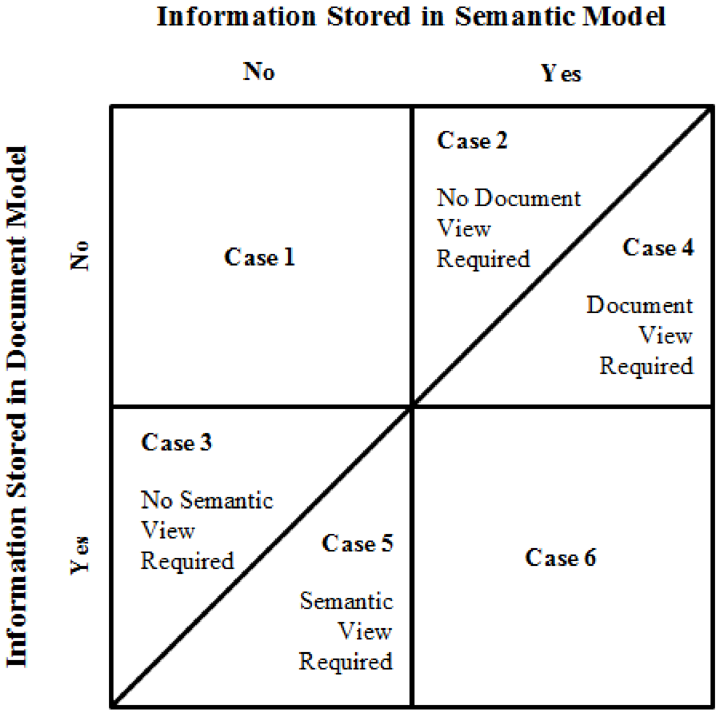

Information Stored in Semantic Model
No
Yes
Information Stored in Document Model
No
Yes
Case 1
Case 2
No Document View Required
Case 4
Document View Required
Case 3
No Semantic View Required
Case 5
Semantic View Required
Case 6

Figure 2. Information Store Matrix

Cases 1-3 of the matrix can be ignored. These cases have an impact on the location of information as, in case 1, the information does not exist and, in cases 2 and 3, information is not required by the alternative source. In theory, cases 4 and 5 are transferable, in that the information can be located in either and then a link or view of the information could be maintained in the other. In these scenarios, case 5 is considered the better option as mentioned previously; the semantic information will only be stored in one model, hence the better option. The only occasion where case 4 is not a limitation is a singular occurrence, in which case this can be transferred into case 5 by relocating the information into the semantic model. This would improve any future development as well.

As case 5 represents the situation where the data can be directly mapped from the semantics to the document view, or at least derived from the semantic content, case 6 presents potentially the most challenging situation. It represents the case where the same information is represented in two mutually exclusive formats such that one format cannot logically be derived from the other. As such, this case presents the need to maintain a relationship between the two equivalent sets in both sources, this would therefore result in no singular, comprehensive primary location, i.e. both locations would be required for the complete picture.

5.3 TDL Semantics

To perform an analysis of the material, knowledge of the semantic model that supports the document prose must also be established. It is beyond the scope of this paper to present the exact models used, but an overview is provided in this section.

At the lowest level of granularity, there exists a Data Dictionary identifying the set of data types defined for use on the link. These types are identified by a unique key, the Data Field

Identifier (DFI) and the Data Use Identifier (DUI) pair, referred to as the DFI/DUI. The set of messages (referred to as J-Messages) that may be transmitted over the link are defined in the form of a Message Catalogue, messages are functionally oriented and contain a number of words, referred to as J-Words, each of which contains a number of fields, the type of which is defined by reference to the relevant item in the Data Dictionary (the DFI/DUI).

Hence, Link 16 messages are tree-structured and must conform to certain well-formed constraints, e.g. all bits in each J-Word must be associated with a DFI/DUI (i.e. all fields must have a defined type). Such constraints have been captured in our models and are described elsewhere [25]. There are a small number of different types of word, and certain elements of the payload are mandated by the word type. This structural set of information covers a large proportion of the standard, covering approximately 4000 pages or some 53% of the document.

In addition, structural modeling has also been undertaken on other sections which detail the network management and Time Division Multiple Access (TDMA) architecture. Once again, there was much information which could be captured and represented by a meta model. Another large section of the standard covers the behavioural aspects of the standard with approximately 2000 pages covering the definition of transmit and receive tables and appendix data defining the behaviour and rules related to message transmission, receipt, storage and display. The appendix rules follow an event (or stimulus) - condition - action type structure. There is a clear process flow within which can be modeled.

5.4 Sample Analysis

Through the analysis of the samples, one key association was observed at an extremely high occurrence rate. It almost exclusively forms entire sections of the samples, specifically the message and data definitions. This association is between an element of text and an instantiated class attribute. In many cases the attribute value of the class is presented directly in the document in the form and format defined in the semantic class. To demonstrate this association consider a semantic model for “Message” defined in Figure 3. Within the document, tables simply render all of a Message’s parts, in a table i.e. Figure 4.

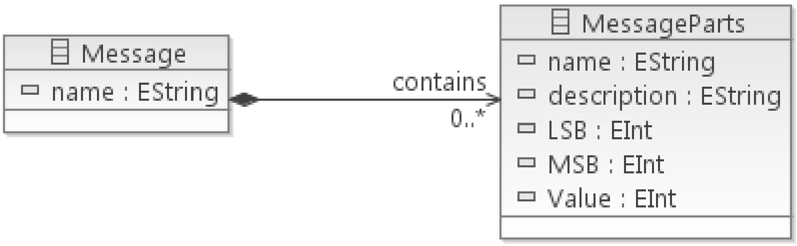
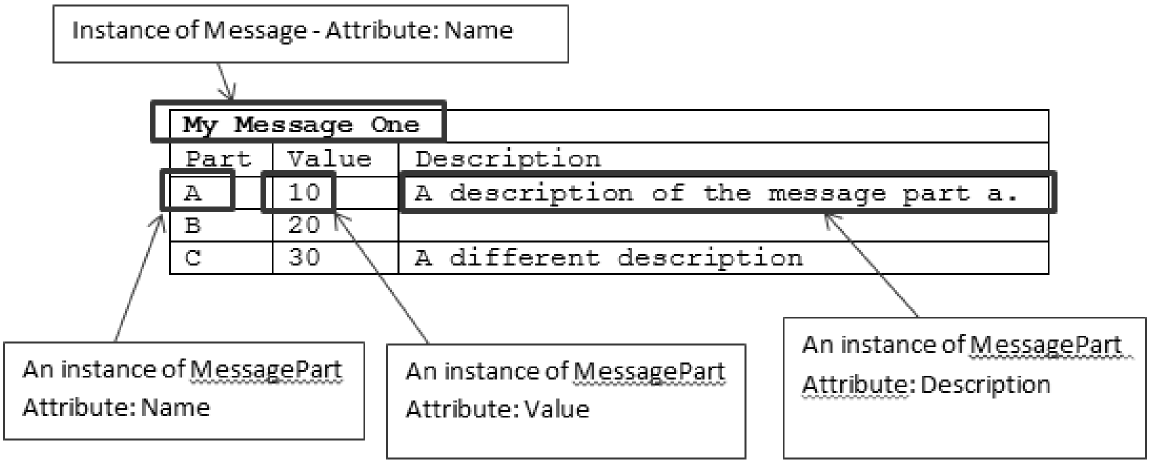


Figure 3. Message Semantic Model



My Message One		
Part	Value	Description
A	10	A description of the message part a.
B	20	
C	30	A different description

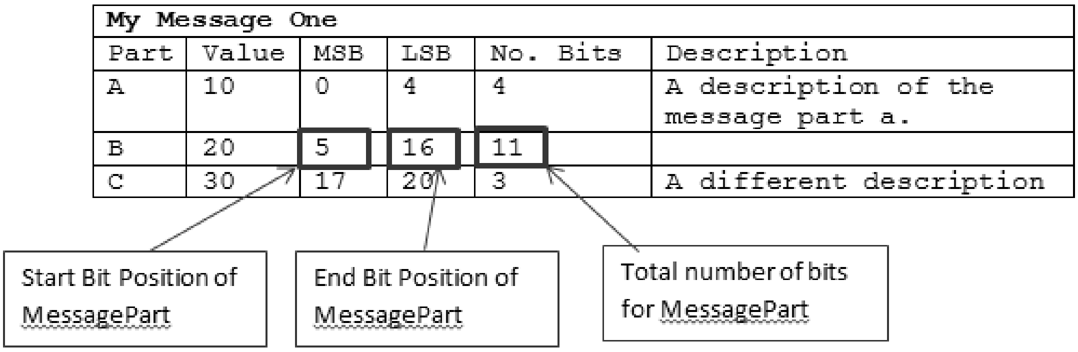
Instance of Message - Attribute: Name
An instance of MessagePart Attribute: Name
An instance of MessagePart Attribute: Value
An instance of MessagePart Attribute: Description

Figure 4. Example Document Content (Message)

To facilitate the aims of hybrid models, a relationship would need to be specified such that this observed association can be modeled. As such, a “direct” relationship should be defined where an instantiated class attribute can be directly utilised from within the document model.

This attribute based substitution can also be observed in other forms. It is hypothesised that the document will contain Boolean information, and the storage format and display format of Boolean data could be different. Most computer based systems use the bit as the storage medium using 0 for false and 1 for true. The hypothesis is that there may exist situations where using a direct replacement of Boolean data will result in illiterate sentence/word structures. Therefore, the bit value should be converted to a more meaningful string for display in the prose, such as, True and False or Yes and No. This could be extended to the consideration of numeric display. In some instances the string can be used but it is the actual numeric value which is stored in the model. Similar synonym-based actions can be observed across the samples. For example, the minimum specification defines a set of characters that are used to indicate mandatory and optional settings. The transmit and receive tables also use characters for the source and message use display type. Therefore, boolean/numeric conversion can be considered as part of the larger concept of synonym substitution, whereby the attribute value is substituted by a synonym. A “synonym” relationship should, therefore, be defined to allow for this reformatting.

A more concealed association was observed in the message and data definitions. Within the samples it could be seen that the text could be derived from the semantic class attributes. Extending the example shown for direct data demonstrates this type of occurrence. For this, we have introduced the start and end bits of the message part. The new table, Figure 5, needs to display the start, end and total bits. Using our semantic model, the number of bits can be derived from the start and end bits. Similarly if it had been modeled with the start and length, the end bit could be derived. Therefore, a “derived” relationship should be defined where the data displayed in the document is a result of the calculation based on attributes on the instantiated class.



My Message One

Part	Value	MSB	LSB	No. Bits	Description
A	10	0	4	4	A description of the message part a.
B	20	5	16	11	
C	30	17	20	3	A different description

Start Bit Position of MessagePart
End Bit Position of MessagePart
Total number of bits for MessagePart

Figure 5. Example Document Usage Semantic Derived Data

The case study document makes use of its highly structured nature to infer further associations. Large sections of narrative are broken into a number of paragraphs. The first paragraph might make reference to a semantic class through the previously discussed “direct” relationship. In doing so there is a clear link established between the paragraph and the semantic data. Any child paragraph can then also infer the relationship through applying the hierarchical structures of a document model. However, the subsequent paragraph, at a sibling level can, in some observed instances, have an implied association with the semantic class referenced in the previous paragraph. The paragraph makes no direct mention or use of data from the semantic class but, through the terminology, spatial proximity and grammar, a human reader can infer the association. Considering an objective of document sub-setting or semantic searching, these sibling paragraphs would be ignored as no relationship would exist. Therefore, some form of “inferred” relationship is required to handle these instances.

During the analysis a common, misleading association was observed multiple times where the document references a table identifier in relation to discussing the semantic class. Because the document model table reference makes use of the semantic attributes from the reference text, it originally was mistaken to be an association to the semantic model. On closer inspection, it is, in fact, a traditional reference within the document model and no hybrid link needs to be established. What it does raise is the association a table has with semantic data. This highlights again the observations of the inferred relationship as the semantic association could possibly be inferred through its child cells or an association would need to be established as per the inferred relationship.

At present the technique for modeling the behaviour within the standard has not yet been implemented at BAE Systems. It is anticipated that further relationships may need to be developed to integrate the behavioural semantics. In particular, it is anticipated that the behaviour may not be directly representable in the document prose. In such cases the observed inferred relationship may be sufficient to handle most cases of behaviour. The main anticipated exception shown in initial behavioural studies implies a need to handle collection associations. Collections are anticipated to handle the situation where a relationship needs to refer to more than one instance of a class. It should be bi-directional in that either a text part is associated with many behavioural objects, or many logically separated text parts define a semantic behavioural object. As such, there should be a “Collection” relationship defined to handle this anticipated requirement.

Therefore, analysis of the sample section has led to the identification of five possible basic relationship types:

- Direct Attribute Substitution
- Synonym Attribute Substitution
- Derived Attribute Substitution
- Inferred Relationships
- Collection Relationships

6 PROPOSED RELATIONSHIP TYPES

Having identified the required relationship types through the analysis of the sample data, the next stage was to propose the relationship definitions to facilitate this. Throughout this process, consideration was given to applying the definition to the material observed in the sampling. A business decision at BAE Systems, meant that the technology for any model would be the Eclipse Modeling Framework. The Eclipse Modeling Framework (EMF) is a modeling framework developed as part of the Eclipse platform [26]. It also provides a code generation facility for building tools and other applications based on a structured data model. The core EMF framework includes a meta model (Ecore [27]) for describing models and runtime support for the models. Therefore, all examples shown and some evaluated proposals are closely coupled to the Eclipse Modeling Framework and the underlying model format Ecore. The Epsilon [28] meta model management family tools are also the favoured technique for model manipulation at BAE Systems.

This section discusses the definitions and the considerations made during application to the observed material. It should be noted that in the examples shown, a class name in italics implies the class is abstract.

6.1 Direct Relationships

Direct Attribute Substitution appears sufficient to cover most occurrences. A direct substitution would simply require replacing the document text with the value of a semantic class attribute. This relationship can be defined by a relationship to the class and an identifier for the attribute of the class as seen in Figure 6.

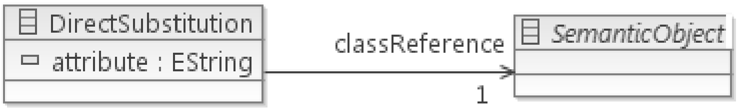


Figure 6. Direct Substitution Proposal

However, a question is raised when applied to the identification of the Message Catalogue and Data Dictionary identifiers. These identifiers appear significantly throughout the semantic information within the samples. For example, DFI/DUIs are identified by the pairing of the

DFI id and DUI id to form a seven to eight character identifier, the J-Messages and their sub-parts are identified by the hierarchical combining of their ids. The J-Message or J-Word precisely demonstrates the conceptual problem. Is the identifier for a J-Word, a singular reference, or should the J-Word be a composition of references where the parent classes are also referenced by the selective substring of the J-Word identifier? To highlight this, Figure 7 shows the example of how the J-Word Z9.3X2 could be composed and represented in the document.

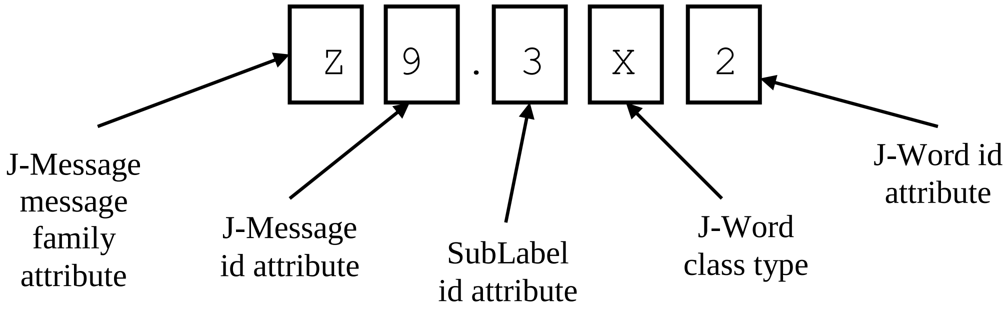


Figure 7. Decomposition of J-Word ID

When referring to the J-Word, the full string reference is used, i.e. “Z9.3X2”. Using simply “X2” would be ambiguous as there are many X2 J-Words of other J-Messages. The critical question is whether instances within the prose are made up of the composition of parts shown in Figure 7, or the full string itself should be the only reference to generate the prose?

By creating all instances within the document as compositions of the hierarchical parts, there is a clear and direct link to each level within the hierarchy. This makes it simple for discovering abstract queries such as finding paragraphs related to high level hierarch objects. However, ignoring our current use of the military standard, considering a scenario whereby the document is being generated through a series of iterations, this composition approach is extremely poor at handling change. For example, consider an iterative step whereby it is decided to move a J-Word X2 or SubLabel Z9.3 to Z9.4 as it was entered incorrectly or due to reorganisation. The use would also need to find all instances within the document of the composite relationships as the reference to the SubLabel would not change and would still be valid. To handle this scenario, it would be better to ensure each hierarchical identifier is a singular reference. By using the object model, relations to the parent hierarchical classes could still be discovered, but the document would not contain any direct knowledge.

To facilitate the singular reference, the relationship needs to have knowledge and ability to traverse the meta model. This traversal is not guaranteed to be fixed depth and the string composition could require differing attributes at each level. Therefore, a programmatic method would be best to generate the necessary text string for the relationship. This method-based substitution relationship, defined in Figure 8, should exist in addition to the direct substitution relationship.

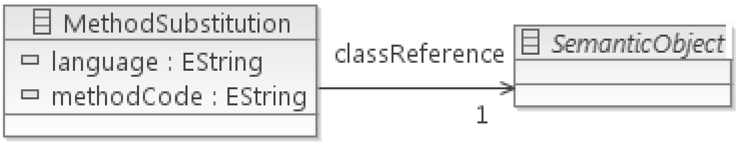


Figure 8. Method Substitution Proposal

Decomposition highlights another potential relationship. Should the referencing be seen as the composition of attributes, then the J-Word type, ('Initial', 'Extension' or 'Continuation') is identified by the inheritance within the semantic class as shown in Figure 9, and not by an attribute of the class. Therefore, a relationship to the class type would be required with the same functionality of synonym such that some text could be used to represent the class type. Although this type of relationship is evident in assessing the composition of the J-Message identifier, there is no other evidence of this elsewhere within the samples. It is, therefore, not deemed necessary to implement this type of relationship. Should it be found that this relationship is required, the functionality could be achieved through the use of the method substitution relationship identified in Figure 8.

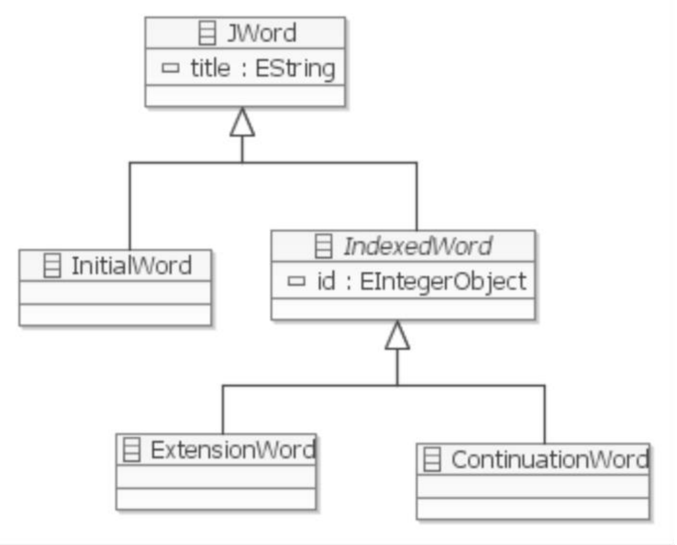


Figure 9. J-Word Inheritance

To provide an example of both direct and method substitution, a common occurrences from within the case study is presented in Figure 10. This represents a paragraph based description for a message which is fictitious for security reasons. This paragraph contains various references to the Message Catalogue. At present, this paragraph is typically captured independently as text in a document model and attributes in the semantic model. The excerpt in Figure 11 shows how the direct and method substitution can be used to markup the document with a reference to the semantic model. The markup has been simplified for readability.

```
"Transmit a X7.0 Message Title 0 message, some text describing low level technical
data and some more text that requires a second new message in the appropriate X3.x
Message Title 1 message or X5.4 Message Title 2 message."
```

Figure 10: Example Paragraph

```
<paragraph><text>Transmit a </text><MethodSubstitution language=”eol”
method=”function buildSubLabelId …” reference="./messages.model/#_IDX7" /><text>
</text><<DirectSubstitution attribute="title"
reference="./messages.model/#_IDX7"/><text> message, some text describing low
level technical data and some more text that requires a second new message in the
appropriate </text><MethodSubstitution language=”eol” method=”function
buildMessageId …” reference="./messages.model/#_IDX3" /><<text>.x
</text><DirectSubstitution attribute="title"
reference="./messages.model/#_IDX3"/><text> message or </text><MethodSubstitution
language=”eol” method=”function buildSublabelId …” /><text>
</text><DirectSubstitution attribute="title"
reference="./messages.model/#_IDX5"reference="./messages.model/#_IDX5"/><text>
message.</text></paragraph>
```

Figure 11 Example paragraph markup including direct & method substituion

6.2 Derived Relationships

The observed samples demonstrate a relatively simple case of derived data where the result is the output of an operation between two attributes. This type of simple derivation can be achieved through a predefined set of operations and links to the necessary attributes. Although this type of implementation would facilitate the examples seen in the samples, it is not highly flexible and is fixed to the set of operations that would be defined in the meta model.

Having established the need for a programmatic method substitution to handle the generation of identifiers in the Analysis Section, this method substitution gives the full flexibility offered by the use of a programmatic language whilst proving all the necessary functionality to achieve a derived substitution. It is proposed, therefore, to drop the derived substitution relationship in favour of using the method substitution relationship. In using the Eclipse Modeling Framework with Epsilon it is proposed that this method substitution is implemented using two techniques. Eclipse has the ability to define eOperations for implementing methods on the class instances. Technique one would be to establish the relation to the result of the defined eOperation. Technique two abstracts out of the Eclipse framework and provides the relation as a text string of code and an implementation language. This would allow us to implement Epsilon as the method language to generate the result. The proposed relationships are defined in Figure 12.

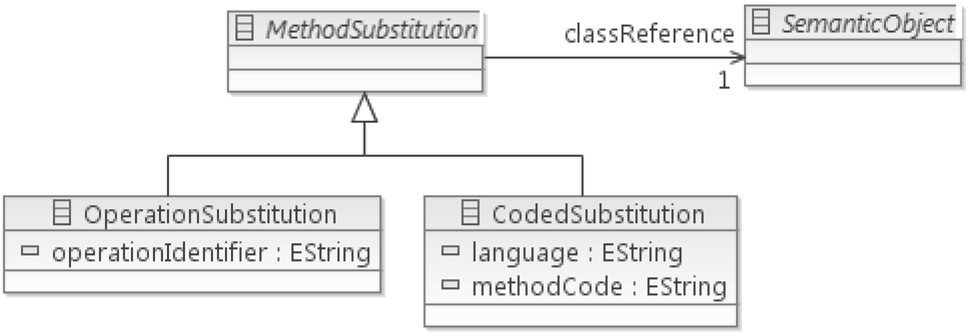


Figure 12. Ecore Modified Method Substitution Proposal

6.3 Synonym Relationships

The Synonym relationship is intended to facilitate the situations where the data held in an instanced semantic class requires formatting into another form. There are three potential methods to achieve this.

1) Basic fixed string replacement.
2) Mapped replacement.
3) Method replacement.

A basic string replacement method would allow the instances of the relation to be assigned a fixed string to be substituted, determined as and when the relation instance is created or modified. Although the basic fixed replacement provides a simple solution for existing documents, similar to that in the discussion of hierarchical identifiers, using a basic fixed replacement is not flexible. Working on the same scenario whereby the document may be created or modified in iterative steps, a fixed replacement suffers from the inability to respond to a change to the semantic attribute value. It would, therefore, be possible to use the method substitution relationship proposed in the derived relationships. This gives complete flexibility but requires a programmatic solution. An alternative would be to use a mapped replacement. A mapped replacement provides a greater flexibility, when defining the relationship; the user must also define a map between values and the required substitution. This method can facilitate the basic fixed method but provides the flexibility to respond to changes to the instantiated semantic class.

The mapped replacement strategy is not immune to faults, consider the scenario where the document author wishes to preserve the language used in the document such that numeric values are represented by their string name i.e. 3 = ‘three’. The map necessary to handle this scenario would need to be infinite to support possible numbers. Although this scenario could be more easily supported through the method substitution, it is also an extreme scenario used solely to highlight the issue. As the mapped approach also enables the map to be defined within the model, this approach is favoured. With the method substitution required for other requirements, this too can be used for any complex mappings. Figure 13 shows our proposed substitution relationship.

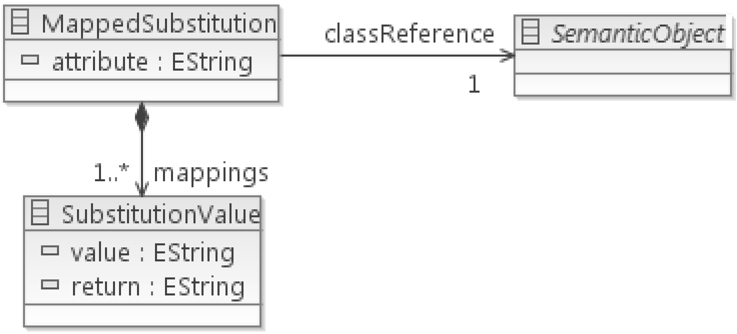


Figure 13. Mapped Substitution Proposal

The mapped substitution is observed mainly in tabular settings. One such example is where an element can be alerted to a user either, textually, graphically or audibly. This is captured in an enumerated type format in the semantic model represented by numeric values. These numeric values should be rendered in the document using a textual representation rather than numeric representation, this is shown in Figure 14.

```
<TableRow><TableCell><MappedSubstitution attribute="title"
reference="./messages.model/#_ID891"><SubstitutionValue Value=”0”
return=”T”/><SubstitutionValue Value=”1” return=”G”/><SubstitutionValue Value=”0”
return=”A”/></MappedSubstitution></TableCell></TableRow>
<TableRow><TableCell><MappedSubstitution attribute="title"
reference="./messages.model/#_ID892"><SubstitutionValue Value=”0”
return=”T”/><SubstitutionValue Value=”1” return=”G”/><SubstitutionValue Value=”0”
return=”A”/></MappedSubstitution></TableCell></TableRow>
<TableRow><TableCell><MappedSubstitution attribute="title"
reference="./messages.model/#_ID893"><SubstitutionValue Value=”0”
return=”T”/><SubstitutionValue Value=”1” return=”G”/><SubstitutionValue Value=”0”
return=”A”/></MappedSubstitution></TableCell></TableRow>
```

Figure 14: Example document table containing mapped substitution

6.4 Inferred Relationships & Collections

The inferred relationship type has no display requirement as it will not be used in a document model to render information. It simply implies an association. As such, the inferred relationship is simply a class with a reference to the semantic class.

Having data returning methods in collection relationships would lead to ambiguity in which element is generating what, and potential synchronisation issues would be created should items be removed, i.e. is the generation still valid? Collection relationships would, therefore, only need to collate inferred relationships. Figure 15 proposes the inferred relationship and the ability to create collection relationships. It facilitates many-to-many style associations through semantic inference and document inference classes.

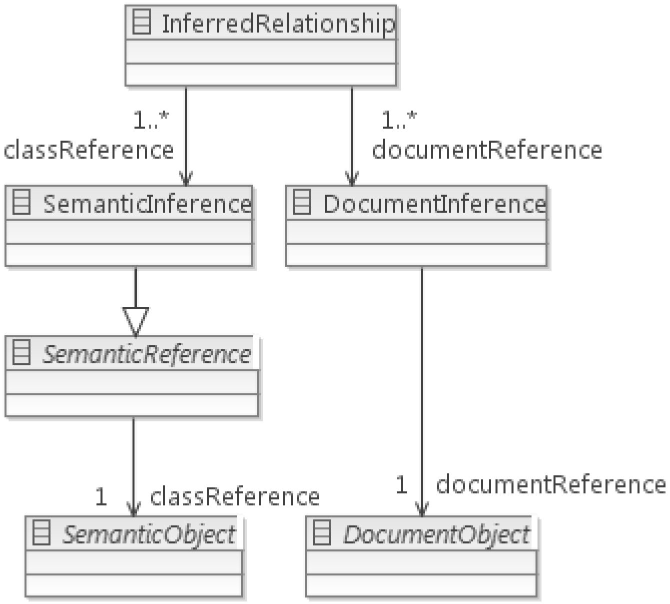


Figure 15. Inferred Relationship Proposal

All of the previously defined relationships are inherently associated with a document model object as they are required for display. The inferred relationship is not and, therefore, needs to create this reference to the document. One method would be to define an object/relationship that resides within the document model, or the semantic model. This would require heavy modification and pre-consideration would be required when defining either model, potentially polluting the actual essence of the model. This would also go against the computing concept of ‘Separation of Concerns’ [29]. By defining a container class such as the Inferred Relationship class in Figure 15, this whole concept can be abstracted out into its own supporting model containing just references to the two other models (like a many-to-many table in relational databases). This ensures that the semantic, document and inferred models stay true to their primary concern, semantics, rendering or mapping, and are not polluted by other concerns.

Within the case study there are paragraphs which describe the user’s actions that occur outside of the system in response to a system message. These paragraphs are related to the message but contain no semantic data more supplementary information. As such inferred relationships could be represented similar to Figure 16.

```
<InferredRelationship><SemanticInference
reference="./messages.model/#_ID891"/><DocumentInferrence
reference="./messages.model/#_PARA1672"/></InferredRelationship>
<InferredRelationship><SemanticInference
reference="./messages.model/#_ID259"/><DocumentInferrence
reference="./messages.model/#_PARA9432"/></InferredRelationship>
```

Figure 16: Example InferredRelationships

6.5 The Semantic & Document Objects

Throughout the definitions discussed above the concept of an abstract Semantic Object is used as a terminal point in all definitions. We propose a simplistic method of achieving this through the definition of the class as an abstract inheritable class. To make a semantic class available to be referenced through our proposed definitions, the semantic model would need to be updated such that the class inherits from this new abstract Semantic Object class. This would allow semantic model owners to also restrict access to elements.

Similarly, the Document Object defined for collection relationships can be achieved in two ways, modifying the base document model such that document model objects also have a reference attribute to the abstract semantic class, or defining an abstract document class that document model elements can inherit from. We favour the construction of an abstract Document Object class that document elements inherit from.

6.6 Final Proposal

Analysing the proposed relationships as a group allows for a level of abstraction which will aid integration with documents. Mapped substitutions can be seen as an extension of direct substitutions as they both require the class attribute. All semantic substitutions have a requirement to relate to a semantic object such that the final proposal defined in Figure 17 defines an abstract class semantic reference which facilitates the abstraction of the semantic association.

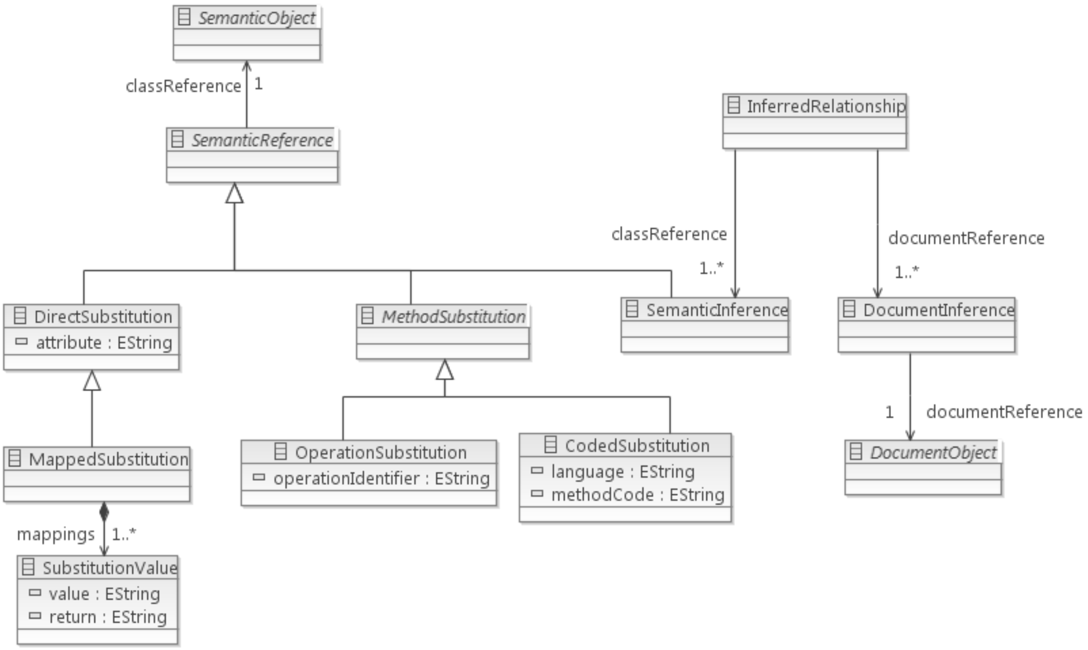


Figure 17. Complete Hybrid Linking Meta-Model

7 INTEGRATION WITH DOCUMENT MODELS

At BAE Systems, for proof of concept and other legacy reasons beyond the scope of discussion in this paper, a custom document model was developed to facilitate the capturing and rendering of the military standard. The document model itself can be found in a previous paper [30]. With the exception of Inferred Relationships, all other relationships are concerned only at the level of a sub-sentence.

A major design decision on defining these relationships was not to tightly couple their implementation to this BAE Systems proprietary document model. Therefore, to incorporate the relationships, they can be used with any document model that, at least, facilitates the breakdown of sentences into a collection of utterances or text parts. Figure 18 demonstrates the integration of the relationship into the BAE System model at the level of text part. In preforming this integration we actually propose another abstract class, Semantic Substitution. This abstract class represents all hybrid relationships that will return a text string, such as substitution relationships. The abstract class makes it easier to factor out a semantic inference as being a viable option as a text part.

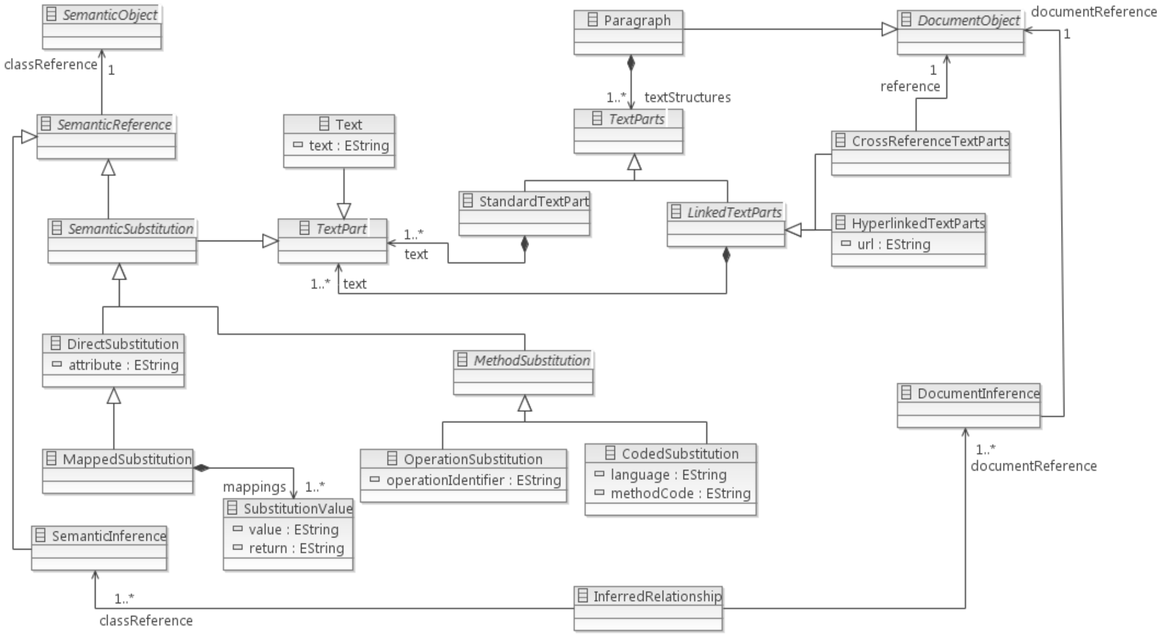


Figure 18. Integrated Hybrid Model

With this proposal, the document model itself contains the semantic relationships (excluding Inferred Relationships). Therefore, the document must contain the instances of the semantic relationships. This raises the philosophical question as to whether the document should contain the hybrid links. One view would be ‘yes’, as all the semantic substitution objects are simply a construct for generating the rendering, which, in essence, is the primary concern of the document model. The other option would be to define a new document model class which is simply a pointer to semantic substitution instances in a separately maintained, hybrid model. The hybrid linking is now separated from the document model, but the document model still needs an element of change to facilitate the reference.

We favour the first view as we consider the purpose of the document model to be the definition of the rendering pattern for the information. As such, the semantic substitution would fall under this concern and be suitably modeled here under the concept of 'Separation of Concerns'. The Inferred Relationships which do not define any rendering can be separated out into another model/package.

As it stands, another key factor in the definition of these models is the intention for them to be utilisable as a bolt-on package to existing document models. This has been designed-in through inheritance. As a semantic model would need to be custom developed for each domain/document, extending this area is not considered a disadvantage. The main requirement here would be extending classes to inherit from the Semantic Object class such that they can be referenced. The document is a potential problem for this bolt-on concept. To reference Document Objects, the document class would need to be modified such that document classes inherit from the Document Object abstract class. Within the BAE System model this is not a problem as the document model already has a concept "Element", which all referenceable document objects inherit, so the Document Object class is replaced by a reference to the actual document model class "Element".

The Document Object is not the only issue, the Semantic Substitution class also needs to inherit from a document class such that is becomes an option when using the document model. The class would need to have no required fields other than a unique identifier and type. In the BAE Systems model, we defined an abstract class "TextPart" from which Text is a child node. We can then make use of this abstract class as the inheritable object for our hybrid meta model. By utilising this abstract class the semantic substitution becomes available as part of building sentences in the document model. Other document models may not have this level of abstraction. If a text part/utterance is defined as a collection of text objects but the text object is left blank, our Semantic Substitution object could inherit from this, maintaining the idea of a bolt-on package. If this is not possible, then another document model would be required or the existing document would need to be 'tweaked' to inject the required abstraction to allow for null text parts.

If these changes are simple, as in the BAE Systems model, then changes are only required to the hybrid meta model i.e. replacing Document Object and inheriting Semantic Substitution. In these cases, the meta model does act as a bolt-on feature where no change is required to the document model. A custom rendering agent would still be required such that the data from these hybrid links can be displayed when trying to render the document.

Therefore the complete requirements for a compatible document model are:

1 It should, at least, facilitate the breakdown of sentences into a collection of utterances or text parts.
2 It has an object that all document elements are inherited from, or it can be modified such that the document class can inherit from a new abstract class.
3 It has a class at the text part level that does not have any required attributes beyond a unique id.

7.1 DITA & DocBook

DITA or Darwin Information Typing Architecture and DocBook are both OASIS standards for technical document structure modeling [31, 32]. DITA, in particular, is highly customisable through the use of specialisations [33]. The keyword object of DITA appears to be a suitable point for our proposed Semantic Substitution to inherit. DocBook has a significant number of elements that facilitate the splitting of a paragraph into smaller elements from which we could inherit. With regards to the Document Object, it is not clear if this could be achieved through reference to existing elements. Both appear to support our idea of facilitating the hybrid models as a bolt-on package from the perspective of Semantic Substitution but more detailed analysis would be required to assess the integration of the inferred relationships.

8 EVALUATION

The proposed solution does achieve many of the anticipated benefits discussed in Section 3. Although some of the benefits could be considered achievable through independent models, the novelty of this approach comes from its coupling to the written document. Capturing the interpretation of complex prose is one such example.

The problem of capturing the interpretation can be solved through the sole use of a semantic model. Using just a semantic model introduces a second source of information and duplication but, by using the relationships defined, elements of the duplication and hence synchronisation issues are reduced. The main benefit of this approach is through the inferred relationships, as sections of semantics can be clearly linked to sections of prose. This will enable novice users of the document to trace a complex paragraph to a formal semantic model. It should help reduce ambiguity through the interpretation of prose by different engineers as there is one formal interpretation underlying it.

The reduction of ambiguity aids the objective of resolving/preventing inconsistencies. By having the relationships, any change to an underlying semantic value is immediately reflected in each instance within the document model. This means changes are not missed by having to go page by page to find reference to the item. This one point of change also meets the objective to make updating information easier. Even the inferred relationships help as, before a change is made, it can be inspected to see where it is utilised within the document and any prose changes can then be made.

It, unfortunately, does not directly improve the situation of resolving missing data, beyond the requirement that the data must exist to be related. The process of generating the required semantic model does aid this, however. In reading the prose to generate the model or starting from a model, it is hoped an engineer would be able to identify any missing elements.

Searching is also only slightly improved. The semantic model can be searched and, given the relationships, the result can then be returned with links to elements within the document model. This does provide new search methods, however it can make searching harder. As the data no longer lives directly within the prose, searches from outside of the rendering application are much harder i.e. a system search within files.

As with the searching, the relationships do provide alternative navigation structures through navigating the semantic model and then tracing the relationships back to sections of the document. The biggest potential of this solution is the objective of performing advanced document comparison. As a document is supported by an underlying semantic model, the comparison can be made at semantic level rather than the textual level. It is not a benefit in its own right, however, as this comparison could be achieved by maintaining a separate semantic model, but the tight coupling gives the appearance to the user that the comparison is operating on the document.

The final objective to discuss is the use of model execution facilities. This objective is only achievable through using the semantic model and can be achieved with or without the use of hybrid models.

In summary, although many of the benefits of this approach can be achieved using independent semantic models and documents, this approach provides benefits by combining them into one. The biggest benefit, compared to maintaining independent document and semantic models, is reducing duplication and synchronisation issues. It also potentially hides the semantics from users with little modeling background, enabling them to utilise the benefits of models whilst maintaining their user friendly document view but having advanced options, such as semantic comparison, available.

9 CONCLUSION

We have presented a framework of relationships which allows for the relationship of objects in a semantic model to be represented in a document such that the information can be rendered without the need to maintain the information both in a document form and a semantic modeled form. This framework facilitates the achievement of many of the benefits anticipated in Section ‘What is the Purpose’. Although much of the benefit can be achieved through using independent models, a significant improvement of this technique is the close coupling which reduces duplication and synchronisation.

9.1 Future Work

This is still an active area of research and there is much that is scheduled for investigation. Of primary importance to the BAE Systems project is a demonstration of the exploitation of the ideas. A proof of concept application which can showcase the anticipated benefits would provide business support for extending and improving on the research.

Another major business driver is behavioural integration, analysing how we can incorporate behavioural modeling, currently under development, into this technique and produce a rendering of this behaviour.

Open source document model integration is another area of interest. Although we have briefly discussed the integration into DITA, if it can be demonstrated that this can easily be applied to the mainstream document models, it has greater benefit for the wider community.

9.2 The Success of the Project so Far

The project has developed an evolving model of substantial elements of the MIL-STD-6016C. The research presented in this paper has identified and successfully defined a set of relationships that allow for the data to be maintained in this semantic model whilst being utilised in our document model. These relationships are defined such that they can be utilised as bolt-on enhancements requiring little modification to the existing document and semantic models.

10 ACKNOWLEDGEMENTS

This paper would not have been possible without the help and assistance provided by Julian Johnson of BAE Systems and Chris Holmes before he left BAE Systems for pastures new. Chris and Julian were involved in the very early work of BAE Systems document and subsequent domain modeling. Julian also acted as the project lead for BAE System.

11 BIBLIOGRAPHY

[1] U.S. Department of Defense, Department of defense interface standard – tactical data link (tdl) 16 – message standard, Tech. Rep. MIL-STD-6016C, US Department of Defense (2004).

[2] NATO, Tactical data exchange – link 16, Tech. Rep. STANAG No. 5516, NATO Standardization Agency (2006).

[3] NATO, Technical characteristics of the multifunctional information distribution systems (mids), Tech. rep., NATO (2001).

[4] C. Holmes, J. Johnson, Tactical data link – domain dictionary, Tech. Rep. SEIC RP 0393, BAE Systems (2005).

[5] B. Zeigler, Simulation-based testing of emerging defense information systems, Presentation [Available at: http://www.acims.arizona.edu/PUBLICATIONS/Presentations/AuburnTalk.ppt] (April 2006).

[6] R. Wieringa, Design science as nested problem solving, in: Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, DESRIST '09, ACM, New York, NY, USA, 2009, pp. 8:1–8:12.

[7] J. Guo, A. Takada, K. Tanaka, J. Sato, M. Suzuki, T. Suzuki, Y. Nakashima, K. Araki, H. Yoshihara, The development of mml (medical markup language) version 3.0 as a medical document exchange format for hl7 messages, Journal of Medical Systems 28 (2004) 523–533.

[8] M. Palmirani, G. Contissa, R. Rubino, Fill the gap in the legal knowledge modelling, in: G. Governatori, J. Hall, A. Paschke (Eds.), Rule Interchange and Applications, Vol. 5858 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2009, pp. 305–314.

[9] G. Antoniol, G. Canfora, G. Casazza, A. De Lucia, E. Merlo, Recovering traceability links between code and documentation, Software Engineering, IEEE Transactions on 28 (10) (2002) 970 – 983.

[10] A. Marcus, J. Maletic, Recovering documentation-to-source-code traceability links using latent semantic indexing, in: Software Engineering, 2003. Proceedings. 25th International Conference on, 2003, pp. 125 – 135.

[11] B. Ramesh, M. Jarke, Toward reference models for requirements traceability, Software Engineering, IEEE Transactions on 27 (1) (2001) 58 –93.

[12] G. Spanoudakis, A. Zisman, E. Pérez-Miñana, P. Krause, Rule-based generation of requirements traceability relations, Journal of Systems and Software 72 (2) (2004) 105 – 127.

[13] S. Winter, S. Nittel, Formal information modelling for standardisation in the spatial domain, International Journal of Geographical Information Science 17 (8) (2003) 721–741.

[14] R. H. Dolin, L. Alschuler, C. Beebe, P. V. Biron, S. L. Boyer, D. Essin, E. Kimber, T. Lincoln, J. E. Mattison, The hl7 clinical document architecture, Journal of the American Medical Informatics Association 8 (6) (2001) 552–569.

[15] R. H. Dolin, L. Alschuler, F. Behlen, P. V. Biron, S. Boyer, D. Essin, L. Harding, T. Lincoln, J. E. Mattison, W. Rishel, et al., Hl7 document patient record architecture: an xml document architecture based on a shared information model., in: Proceedings of the AMIA Symposium, American Medical Informatics Association, 1999, p. 52.

[16] R. H. Dolin, L. Alschuler, S. Boyer, C. Beebe, F. M. Behlen, P. V. Biron, A. S. Shvo, Hl7 clinical document architecture, release 2, Journal of the American Medical Informatics Association 13 (1) (2006) 30–39.

[17] R. H. Dolin, L. Alschuler, Approaching semantic interoperability in health level seven, Journal of the American Medical Informatics Association 18 (1) (2011) 99–103.

[18] A. Hasman, et al., Hl7 rim: an incoherent standard, in: Ubiquity: Technologies for Better Health in Aging Societies: Proceedings of MIE2006, Vol. 124, Ios PressInc, 2006, p. 133.

[19] C. N. M. Gunther SCHADOW, The hl7 reference information model under scrutiny, in: Ubiquity: technologies for better health in aging societies: proceedings of MIE2006, Vol. 124, Ios PressInc, 2006, p. 151.

[20] R. L. Baskerville, A. Wood-Harper, A critical perspective on action research as a method for information systems research, Journal of Information Technology 11 (3) (1996) 235–246.

[21] I. Benbasat, D. K. Goldstein, M. Mead, The Case Research Strategy in Studies of Information Systems, MIS Quarterly 11 (3) (1987) 369–386.

[22] R. Goede, C. de Villiers., The applicability of grounded theory as research methodology in studies on the use of methodologies in is practices., in: SAICSIT ’03, 2003, pp. 208–217.

[23] J. Iivari, A paradigmatic analysis of comtemporary schools of is development, European Journal of Information Systems 1 (4) (1991) 249–272.

[24] A. Salminen, K. Kauppinen, M. Lehtovaara, Towards a methodology for document analysis, Journal of the American Society for Information Science 48 (7) (1997) 644–655.

[25] C. Holmes, J. Johnson, A. Riaz, Tactical data links – a description of the supporting models, Tech. Rep. SEIC-RP-0531, BAE Systems (2007).

[26] D. Steinberg, F. Budinsky, M. Paternostro, E. Merks, EMF: Eclipse Modelling Framework, 2nd Edition, Eclipse Series, Addison-Wesley Professional, 2008.

[27] IBM and others, org.eclipse.emf.ecore (emf javadoc), this is an electronic document. Date of publication: Jun 23, 2010. Date retrieved: Nov 11, 2010 (2010).

[28] D. S. Kolovos, R. F.Paige, F. A. Polack, The epsilon object language, in: Proc. European Conference in Model Driven Architecture (EC-MDA) 2006, Vol. 4066 of LNCS, Bilbao, Spain, 2006, pp. 128–142.

[29] W. L. Hürsch, C. V. Lopes, Separation of concerns, Tech. rep., College of Computer Science, Northeastern University (1995).

[30] D. Clowes, R. Dawson, S. Probets, J. Johnson, C. Holmes, Pilot studies in using the semantic knowledge of information in large technical documents to aid user navigation, in: D. Harorimana, D. Watkins (Eds.), The 9th European Conference on Knowledge Management: Eckm 2008, Academic publishing, 2008, pp. 129–138.

[31] OASIS, The docbook schema version 5.0 (November 2009).

[32] OASIS, Darwin information typing architecture (dita) version 1.2 (December 2010).

[33] E. Hennum, Specializing domains in dita, Tech. rep., IBM (September 2005).

Darren Clowes is an Engineering Doctorate student at Loughborough University. He received his masters degree in Computer Science at Loughborough University and recently has been working at the Systems Engineering Innovation Centre in collaboration with BAE Systems on standards modeling and exploitation.

Ray Dawson is Professor of Knowledge Management at Loughborough University, UK. He obtained his bachelor's and masters degrees from Nottingham University before entering Plessey Telecommunications in 1977. At the company he developed an interest in software engineering processes and the working methods for information systems development as practiced in industry. Since 1987 he has continued this interest in industrial working methods at Loughborough University, working with companies to improve their software processes and information systems. Ray Dawson is now Deputy Head of the Computer Science Department and Associate Dean for Teaching of the School of Science at Loughborough University.

Steve Probets is a lecturer who has recently moved from the Department of Information Science to the Department of Computer Science at Loughborough University. He is programme tutor for the Department’s ‘Information Management and Computing’ programme. His research interests include electronic publishing and document engineering and he has supervised a number of PhD candidates looking into mechanisms to improve the usability of both electronic documents and websites. Other research interests include the evolution of scholarly publishing.

Appendix B ECMFA Conference Paper

Paper published in 2010 in Modelling Foundations and Applications, volume 6138 of Lecture Notes in Computer Science, pp. 62–73.

A Reflective Approach to Model Driven Web Engineering

Darren Clowes1,2
Dimitris Kolovos3
Chris Holmes2
Louis Rose3
Richard Paige3
Julian Johnson2
Ray Dawson1
Steve Probets4

Received: date / Revised version: date

1, Department of Computer Science, Loughborough University, Leicestershire, UK.
2,BAE Systems, SEIC, Loughborough University, Leicestershire, UK.
3, Department of Computer Science, The University of York, Heslington, York, UK.
4, Department of Information Science, Loughborough University, Leicestershire, UK.

Abstract

A reflective approach to model driven web engineering is presented, which aims to overcome several of the shortcomings of existing generative approaches. The approach uses the Epsilon platform and Apache Tomcat to render dynamic HTML content using Epsilon Generation Language templates. This enables EMF-based models to be used as data sources without the need to pre-generate any HTML or dynamic script, or duplicate the contents into a database. The paper reports on our experimental results in using this approach for dynamically querying and visualising a very large military standard.

1 INTRODUCTION

Increasingly, the design of complex engineered products and systems is becoming more reliant on computer-supported models, capturing structured information. By contrast, most military standards in use are still disseminated as text-based documents. In our experience, this is also the case in other domains requiring complex detailed standards such as automotive and aeronautical industries. These standards can be over thousands of pages in volume, which, can make locating and composing information in them challenging and laborious. Within the development of military hardware, engineers are typically required to produce documents based on a subset of the standards implemented by their product. Product testing will require the engineers to promptly locate information in the standard and validate that their product conforms to it. Interoperability is also crucial in military applications where engineers must validate that their implementation does not impede or conflict with other products. To address these issues, we have found it useful to extract semantic models represented in a structured format which is then amenable to automated querying and processing. In our experience this approach can greatly enhance both the accuracy and speed of locating and composing information from different parts of a standard. We have also found that in order to make models useful for engineers, it is essential to construct a suitable and familiar user interface for querying and navigating them.

In our previous work we used text parsing to extract structured Eclipse Modelling Framework(EMF) based models from text-based military standards. This paper presents our work on using a combination of web and Model Driven Engineering (MDE) technologies (in particular the Epsilon Generation Language (EGL) and Apache Tomcat) to enable dynamic querying and visualising of these models over the web. The rest of the paper is organized as follows. In section 2 we provide an overview of the domain of military standards with an emphasis on Tactical Data Links – which is the main focus of our work. Then, in section 3 we outline the motivation for querying and visualizing models over the web. In section 4 we perform a review of existing generative MDE approaches for implementing web-based applications and highlight their advantages and shortcomings. Driven by the findings of this review, in section 5 we propose a novel reflective approach for building web-based applications directly atop EMF-based models. Then, in section 6 we evaluate this approach both from a development effort and a performance perspective and assess its suitability for building real-world applications. In section 7 we conclude and provide directions to envisioned future work.

2 BACKGROUND

In this section we provide an overview of the military standards on which our work focuses (Tactical Data Links) and outline our previous work on extracting EMF-based models from text-based standards documents.

2.1 An Introduction to Tactical Data Links (TDLs)

The TDL provides one of the backbone technologies underpinning the defence community’s goal of network enabled capability by providing the information and infrastructure to afford users with an integrated picture of the battlefield. It also supports tasking orders and responses. A number of TDLs are in service with coalition forces, and are implemented on a variety of assets, such as aircraft, ships, land vehicles, and command stations.

The Link 16 TDL is described by the Military Standard MIL-STD-6016C [1] in the form of narrative combined with many tables and relatively few figures. At the lowest level of granularity exists a Data Dictionary identifying the set of types defined for use on the link. These types are identified by a unique key the Data Field Identifier (DFI) and Data Use Identifier (DUI) pair, referred to as the DFI/DUI. The set of messages that may be transmitted over the link are defined in the form of a Message Catalogue. Messages are functionally-orientated and contain a number of words, each of which contains a number of fields, the type of which is defined by reference to the relevant item in the Data Dictionary (the DFI/DUI). Hence, Link 16 messages are tree-structured and must conform to certain well formed constraints, e.g. all bits in each J-Word must be associated to a DFI/DUI (i.e. all fields must have a defined type); such constraints have been captured in our models and are described elsewhere [2]. There are a small number of different types of word, and certain elements of the payload are mandated by the word type.

The description of Link 16 provided by MIL-STD-6016C [1] is known to feature a number of shortcomings affecting its usability [3]. The following are of particular relevance to the research reported in this paper:

- Document-based, no apparent underpinning model
- Largely narrative
- Open to (mis)interpretation
- Not checkable by machine
- Duplication of material invites inconsistency
- Only very limited use of hyperlinking impedes document navigability
- Comprises many interdependent sections and appendices
- Size, >7300 pages

The description of the Data Dictionary and Message Catalogue components comprises approximately 4000 pages of structured text, a vast majority of which does not feature hyperlinking; bookmarks are only provided in the PDF version of the standard but at a relatively coarse level of granularity. As a result, locating information within this document is a particularly challenging and tedious task for engineers.

2.2 Modelling TDLs

An analysis of the Link 16 TDL standard [1] lead us to the conclusion that the domain can be effectively captured by a hierarchically layered set of metamodels, the lower two layers of which comprise the Data Dictionary and Message Catalogue; we refer to this hierarchy of models colloquially as the semantic models. An excerpt of the Message Catalogue can be observed in Fig. 1. Modelling commenced in mid 2005 using Xactium's XMF Mosaic tool. Full scale development and support of the XMF tool ended in 2008. As such, in early 2009 we began migrating some model components into EMF, and the Epsilon framework [4] as part of a risk reduction exercise investigating candidate successor tools for XMF. EMF provides automated metamodel generation from an XML schema, and, as much of our source data is available in XML, generated via bespoke parsers we have written for the project, migration of the Data Dictionary and Message Catalogue proved to be feasible. The Epsilon Transformation Language (ETL) was used to transform the XML data, conforming to the automatically-generated metamodels, to conform to our derived *semantic models*.

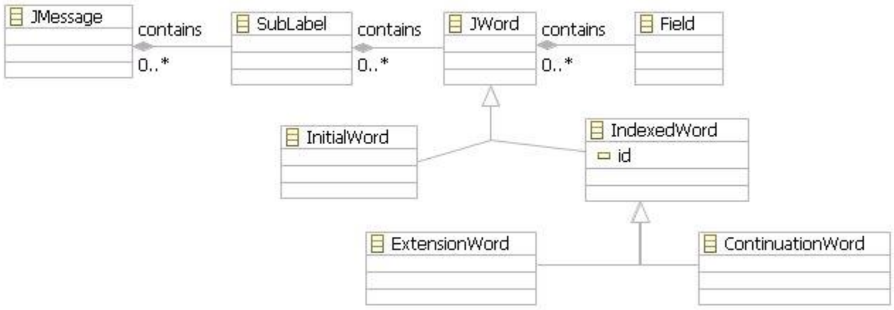


Figure 1: Message Catalogue Excerpt

3 MOTIVATION

Migrating the standard to a model-based form provides an opportunity to address many of the shortcomings identified against the current document-based view. However, for a model-based representation of the standard is to be accepted by TDL practitioners we must be able to provide access to the data in these models in a format similar to that of the original document, but based on a sound foundation, validated against the relevant well formed constraints and with enhanced capabilities for document navigability. Analysis in 2007 investigated the navigation methods available to engineers and the benefits that could be gained from a model based approach [5].

Engineers use the standard in every stage of a product's lifecycle. Some key activities include using the standard to investigate and confirm message implementations, developing regulatory documents based on subsets of the standard, and reviewing interpretations and interoperability with other products. Therefore the information in the standard needs to be visualised with usable and familiar interfaces. The engineers are typically used to navigating the information in the traditional document view of the PDF. The document also contains much explanatory prose which cannot be modelled, but which nevertheless is essential for the users understanding. Methods to quickly traverse the Message Catalogue and Data Dictionary with the ability to move between cross-references will also potentially improve an engineer's efficiency in using the standard.

Hence, it is necessary for the underpinning *semantic models* to be rendered in a text-based but cross-linked manner. Developing visualisations of the modelled data in HTML has been deemed as a preferred solution for the current stage of development at which the TDL modelling research is at. Using HTML also means that deployment to TDL engineers can be easily achieved through the current facilities available to them, i.e. the web browser of their desktop machines. Developing a desktop-based application has the disadvantages of requiring installation and security validation for each project/engineer that wishes to use this functionality. Also, as the models are still evolving new development can simply extend the web application rather than requiring newer versions of the desktop-based application to be installed.

This motivation to render the model data and the desire to utilise HTML as the deployment technology has led to the need to investigate the different options for creating web applications based on our EMF models of the TDL Link 16.

4 RELATED WORK: MODEL DRIVEN WEB ENGINEERING

The last decade has seen a growth in the adoption of Model Driven Web Engineering (MDWE). MDWE aims at applying MDE principles to web application development. MDE advocates the use of models and model transformations as first-class artefacts in all phases of software development, and promotes the abstraction of models to be platform independent, with subsequent transformations to generate platform specific models for deployment. Many MDE methods and tools also enable the automated generation of application code from these models. MDE has the potential to greatly reduce development and maintenance costs, while increasing the quality of the software produced. There are many model driven web engineering methods, some prominent examples include, OO-H [6], UML-based Web Engineering (UWE) [7], WebML [8], WEI [9] amongst others.

Nearly all methods can be considered to consist of three platform independent models. These can be generalised to be, a concept model, a navigation model and a presentation model. Methods of transformations to platform specific models vary from use of graph transformations like MIDAS [10] to template based like WebML [8]. Predominantly QVT and ATL are used by these approaches for the transformation between models. In addition, constraints are generally written using OCL. Several solutions make use of their own languages or language extensions. For example WebML utilises its own extensions of UML, OO-H uses Navigation Access Diagrams, WEI defines a custom toolset (GlueWeb), which is, an incomplete subset of OCL combined with QVT [9].

All model driven web engineering approaches currently take a generative approach to producing the final web application by generating platform-specific code from the respective platform-independent models. For dynamic application, this includes the production of a data store either through generating a relational database based on the models, or through transformations to the Ecore XMI and utilising the data through XML. Some approaches do make use of the EMF models through generating the Java model code and model editor classes. The presentation coding is generated by all approaches whether the output is PHP, JSP or static HTML. In our view, generative approaches demonstrate several disadvantages. Any change to the metamodels requires the regeneration and redeployment of the code. Also, data is duplicated and this leaves open the potential for synchronisation issues. Moreover, as the amount of data in the models grows, re-generating the entire contents of the database every time a model changes can be particularly time-consuming. In addition, there can be an impedance mismatch between the object-oriented metamodels and relational storage solutions. As our *semantic models* are evolving, they are increasingly making use of more complex object-oriented features, such as multiple inheritance, therefore, the corresponding changes in the relational view and the necessary mappings are becoming more challenging. However, the level of abstraction used, means that transformations are still performed at the ecore model level and knowledge of the relational database structure no longer a problem. There is still the execution overhead of transformations as the data grows.

5 A REFLECTIVE APPROACH TO MDWE

To overcome the shortcomings of existing generative approaches to web development, and particularly data duplication and re-generation, we decided to investigate the feasibility of an alternative, reflective approach in which we could use the EMF-based models themselves – instead of duplicating their contents in a database – as the data source atop which we would build the web application. Moreover, we decided to investigate the possibility of using the Epsilon Generation Language (EGL) to express the templates that would generate dynamic HTML content from the underlying EMF model.

5.1 Technical Infrastructure

In this section we outline the technical details of our approach. We first introduce EGL and its underpinning Epsilon platform and then discuss integrating EGL with a Java-based Web Server (Apache Tomcat) that allows us to implement reflective web-applications using EMF-based models as data sources, and EGL templates for querying and producing dynamic HTML content from them.

5.1.1 Epsilon and EGL.

Epsilon is a component of the Eclipse Modelling GMT project that provides tools and domain-specific languages for Model Driven Engineering. Epsilon comprises a number of integrated model management languages, based upon a common infrastructure, for performing tasks such as model transformation, comparison, merging, in-place transformation, inter/intra-model consistency checking, and model to text transformation. All languages in Epsilon build atop the Epsilon Object Language (EOL), an OCL-based imperative model navigation and modification language, and can be used to manage models expressed in different technologies such as EMF, MDR and XML.

Epsilon Object Language (EOL).

EOL – the core language of Epsilon – combines the procedural style of scripting languages such as Javascript with the declarative style of OCL for querying and filtering collections. EOL is a mature language that boasts a wide range of features [11] such as support for managing multiple models of arbitrary modelling technologies in the context of the same program, tight integration with Java enabling developers to instantiate Java objects and call their methods from EOL, support for defining operations in the context of existing types, reuse facilities for defining and importing libraries of operations, support for user-interactions and support for transactional management of models (where the underlying modelling technologies provides such capabilities).

Epsilon Generation Language (EGL).

EGL is a template-based language that targets model-to-text transformation [12]. EGL adopts a syntax that closely resembles server-side scripting languages such as JSP and PHP. An EGL template consists of two types of regions. Dynamic regions (enclosed within [% %]) contain executable statements and expressions, while static regions contain plain text that is output verbatim. For example, consider the simple EGL template in Fig. 2.

```
[%for (i : Integer in Sequence{1..3}){%]
Number [%=i%]
[%}%]
```

Figure 2: Example EGL template

EGL is a preprocessed language; EGL templates are transformed to EOL programs (similarly to the way JSP pages are transformed to Java servlets) which are then executed in order to produce the output. By building on top of EOL, EGL inherits the rich set of features that EOL provides and which were outlined above. In addition, EGL provides a range of task-specific features such as support for dynamic template instantiation and invocation, and support for mixing manually written code with generated code through a target-language independent content preservation mechanism.

While EGL was originally developed to support code generation, its modular design makes it possible to use it to produce text in non-file output streams as well. In principle any model-to-text transformation language with similar characteristics such as XPand, MOFScript or the OMG M2T could have been used instead.

5.1.2 Tomcat.

As discussed before, the aim of this work was to implement a solution that would allow engineers to explore EMF models through standard web browsers. To achieve this, we have implemented an integration between Apache Tomcat and EGL, which allows developers to use EGL templates as server-side pages for rendering EMF models over HTML. This section discusses the rationale and architecture of this approach as well as some of the interesting implementation challenges encountered.

5.1.3 Rationale.

Tomcat is an industrial strength, Java-based web server with built-in support for the JSP server-side scripting language. Therefore, our first option was to use JSP in order to produce dynamic HTML pages from our EMF models. Since EMF is a Java-based library, this capability was available out-of-the-box. However, there was a major disadvantage to this. If we were to use JSP for this purpose, we would need to either generate Java code from our Ecore metamodel or navigate our models using the cumbersome reflective syntax of EMF. By contrast, integrating Tomcat with EGL would allow us to use the concise closure-based syntax provided by EGL[12] to query models, without needing to generate and deploy code for the respective Ecore metamodels in Java.

5.1.4 Architecture.

Although Tomcat comes with built-in support for JSP, like the majority of web servers, it also provides a flexible model for integrating additional server-side languages. This is achieved through the URL mapping mechanism which allows developers to map request URLs to custom servlets. In the case of EGL, we implemented an EGL servlet which is responsible for serving calls to EGL pages and mapped it to requests which end with .egl as shown in Fig. 3.

```
<servlet>
  <servlet-name>egl</servlet-name>
  <servlet-class>org.eclipse.epsilon.egl.servlet.EglServlet
  </servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>egl</servlet-name>
  <url-pattern>*.egl</url-pattern>
</servlet-mapping>
```

Figure 3: Tomcat EGL Servlet mapping

Once the EGL servlet is invoked as a result of a client (browser) request, it is responsible for locating the respective EGL template for each request, executing it and returning the produced text to the client. Similar to JSP pages, EGL templates can access several built-in variables such as the *request* variable which allows a template to retrieve information related to the particular request (e.g. parameters), the *session* variable which allows templates to query and set session-wide properties (e.g. for authentication), and the *response, config* and *application* variables. These variables are inherited directly by the Java servlet specification[13]. To interact with EMF models, each EGL template is provided with a shared instance of the *ModelManager* class which provides operations for loading, storing and disposing of EMF models [14].

5.2 Technical Solution

Using the Tomcat/EGL integration, three different templates were developed in an alpha test application. These templates covered three distinct areas of the Link 16 TDL modelling work. The areas covered were the Data Dictionary, Message Catalogue and the prose document. The prose document template regenerated a rendering of a subset of the standard in the same view style as the PDF format. This template was designed to test scalability as it utilised over 900,000 instances of classes.

The Data Dictionary and Message Catalogue were intended to allow users to traverse their hierarchical structures and follow any cross-references between them. Navigation was provided by using hyperlinks and specifying new parameters using the HTML GET method of the URL. By providing parameters, this enables the template to restrict the data and traverse the hierarchy. An example of this is shown in Fig. 4. This example is returning all child instances conforming to a specific type (InitialWord, ContinuationWord or ExtensionWord), as defined by the passed parameter (`wordType`). The EOL select operation returns a set where all instances conform to the type and the instances id matches the additional parameter (`wordID`) that is supplied. The excerpt also corresponds to the metamodel excerpt seen in Fig. 1.

```
var wordType := request.getParameter('jword1');
if(wordType = 'I')
{x := y.contains.select(t|t.isTypeOf(InitialWord)).first();}
else if (wordType = 'C')
{x := y.contains.select(t|t.isTypeOf(ContinuationWord) and
                        t.id = wordID.asInteger()).first();}
else if (wordType = 'E')
{x := y.contains.select(t|t.isTypeOf(ExtensionWord) and
                        t.id = wordID.asInteger()).first();}
```

Figure 4: Excerpt of selecting element defined by passed parameters

Having determined the set or instance to display, a custom print operation is called to render the data. The custom print operations (e.g. `x.print();`), produce the HTML code for displaying the instance attributes to the screen. In Fig. 5, a short excerpt is given for rendering the top level class JMessage of the Message Catalogue. This will result in a HTML table displaying the data associated to a JMessage.

```
operation JMessage print() {%]
<table>
 <tr><td>Name:</td><td>[%=self.name%]</td></tr>
 <tr><td>Family:</td><td>[%=self.family%]</td></tr>
 <tr><td>ID:</td><td>[%=self.id%]</td></tr>
 ...
```

Figure 5: Excerpt of a custom print operation

It is envisaged that engineers should be able to utilise the modelled data to produce required regulatory material according to subsets of the standard. To achieve this, the solution is required to be able to create and/or edit the data. Fig. 6 demonstrates the ability to select and edit the data of a JMessage.

```
var editme := c.contains.select(x|x.id = msgID and x.family =
                                msgFamily).first();
if(editme.size() == 1){
  editme.name = msgName;
  editme.family = msgFamily;
  editme.message = msgID;
  myCatalogue.store();
}
```

Figure 6: Example of editing a message

6 EVALUATION

The development process is evaluated with respect to its value to model driven web engineering, before considering in more detail the performance of our approach.

6.1 Development Process

Our approach has demonstrated that reflective model driven web engineering is possible. It improves on generative approaches by reducing the amount of duplicated data and therefore reducing the risks and problems associated with synchronisation. By using the Epsilon platform, a range of model management functions can be developed using the common syntax provided by EOL, which, allows developers to reuse code across the different Epsilon languages. Reuse of code is not possible in most generative approaches, as their model management functions utilise differing languages with no common syntax. EOL is also beneficial over OCL as it combines the procedural style of scripting languages such as JavaScript with the declarative style of OCL for querying and filtering collections. Although EOL is a new language, the learning of this is not considered an issue. The syntax is similar to JavaScript and it can be seen as a benefit. Engineers with no modelling experience are not required to learn multiple languages for model management functions such as, transformation languages QCT or ATL or constraint languages like OCL. No technical knowledge of dynamic web scripts or languages is required, this reduces the required knowledge of a modelling engineer to produce a dynamic web based application.

However, the approach does shift away from model driven engineering principles slightly. Most generative approaches utilise a model and subsequent transformations to generate the dynamic script. Our approach currently ignores this model driven approach to the development of the interfaces in favour of a programmatic style that utilises the model driven functions available through the Epsilon platform.

6.2 Performance

Stress and load testing of the alpha test application resulted in an acceptable level of performance for the anticipated usage at BAE Systems. As TDL is a specialised area, it is not envisaged for more than 10 users to use the application concurrently. The stress test results shown in Table 1 show that this application provides adequate performance for 50 users. Studies suggest a web user is willing to accept an 8-10 second delay in loading a page [15, 16]. However, there is some supporting research that suggests that under some conditions 30 seconds is an upper limit [16, 17]. Therefore, the standard deviation coupled with the average load time for 100 users is likely to be unacceptable.

Table 1: Stress Test Results (Users per Minute

		Load Time (ms)					
Users	Requests	Min	Max	Avg	∂	Avg File Size (B)	Failure Rate
10	1565	45	13058	1110	2650.502	16362	0.3%
25	3711	55	30262	2649	6217.821	17489	0.3%

50	8057	52	69979	5041	13194.568	15922	0.5%
100	15426	77	130983	11184	27084.301	16715	0.8%

The results do demonstrate that the approach is not suitable for request intensive applications such as Google or Amazon as the weight of requests would degrade performance significantly. Load testing also discovered a performance bottleneck during the first query of a template. This overhead was observed as the models were loaded in the java virtual machine memory. For large models like the prose document, this can take in excess of five minutes. However, subsequent calls to the templates do not suffer as the models are already in memory.

All the tests were performed using Tomcat and the Java Virtual Machine in their default setting with the exception of increased heap size. Performance enhancements to Tomcat [18] and the virtual machine, are considered to result in only a slight improved performance.

7 CONCLUSIONS

The alpha test application discussed in Sec. 5.2 has demonstrated that the approach of using Epsilon Generation Language with Apache Tomcat is a plausible solution. It provides a reflective template driven method to producing dynamic web content from EMF models. By using the Epsilon framework, dynamic web scripts can be produced which require no generation of supplementary code. This reduces the amount of duplicated code. Whilst Epsilon is a new language to learn, it provides a benefit in providing a range of model management functions that utilise the same core EOL language.

The performance testing in Sec. 6.2 has shown the solution is suitable when using small to medium models with a small number of concurrent users (<50), such as the scenario discussed at BAE Systems. For large scale models and high concurrent request applications the performance degrades to unacceptable levels. This performance degradation is an issue for ongoing further research.

REFERENCES

1 U.S. Department of Defense. Department of defense interface standard – tactical data link (tdl) 16 – message standard. Technical Report MIL-STD-6016C, US Department of Defense, 2004.

2 C. Holmes, J. Johnson, and A. Riaz. Tactical data links – a description of the supporting models. Technical Report SEIC-RP-0531, BAE Systems, 2007.

3 B. Zeigler. Simulation-based testing of emerging defense information systems. Presentation [Available at: http://tinyurl.com/ycwfdbz], April 2006.

4 Eclipse Foundation. Epsilon eclipse modeling gmt component. http://www.eclipse.org/gmt/epsilon.

5 D. Clowes, R. Dawson, S. Probets, J. Johnson, and C. Holmes. Pilot studies in using the semantic knowledge of information in large technical documents to aid user navigation. In *Proc. European Conference on Knowledge Management 2008*, 2008.

6 J. Gómez and C. Cachero. *Information modeling for Internet applications*, chapter OO-H Method: extending UML to model web interfaces, pages 144–173. Idea Group Inc (IGI), 2002.

7 N. Koch. *Software Engineering for Adaptive Hypermedia Systems: Reference Model, Modeling Techniques and Development Process.* PhD thesis, LMU Munich, 2000.

8 S. Ceri, P. Fraternali, and A. Bongio. Web modeling language (webml): a modeling language for designing web sites. *Computer Networks*, 33(1-6):137 – 157, 2000.

9 N. Moreno and A. Vallecillo. Towards interoperable web engineering methods. *Journal of the American Society for Information Science and Technology*, 59(7):1073 – 1092, 2008.

10 P. Cáceres, E. Marcos, B. Vela, and R. Juan. A mda-based approach for web information system. In *Development, Proceedings of Workshop in Software Model Engineering.*, 2004.

11 D.S. Kolovos, R.F.Paige, and F.A.C. Polack. The epsilon object language. In *Proc. European Conference in Model Driven Architecture (EC-MDA) 2006*, volume 4066 of *LNCS*, pages 128–142, Bilbao, Spain, July 2006.

12 L.M. Rose, R.F. Paige, D.S. Kolovos, and F.A.C. Polack. The epsilon generation language (egl). In *Proc. European Conference in Model Driven Architecture (ECMDA)*, 2008.

13 Sun Microsystems. Jsr-000315 java servlet 3.0 specification, January 2009. This is an electronic document. Date retrieved: June 9, 2009.

14 Epsilon Eclipse Modeling GMT component. Using egl as a server-side scripting language in tomcat. http://www.eclipse.org/gmt/epsilon/doc/articles/egl-server-side/.

15 J. Palmer. Designing for web site usability. *IEEE Computer*, 35:102–103, 2002.

16 F.F. Nah. A study on tolerable waiting time: how long are web users willing to wait? *Behaviour & Information Technology*, 23(3):153–163, 2004.

17 P.R. Selvidge, B.S. Chaparro, and G.T. Bender. The world wide wait: Effects of delays on user performance. *International journal of Industrial Ergonomics*, 29:15–20, 2002.

18 V. Chopra, S. Li, and J. Genender. *Professional Apache Tomcat 6*, chapter Performance Tuning, pages 561–584. Wrox Publishing Inc, 2007.

Appendix C ECKM Conference Paper

Paper published in 2008 in 'The 9th European Conference on Knowledge Management: Eckm 2008, pp. 129–138.

Pilot Studies in Using the Semantic Knowledge of Information in Large Technical Documents to Aid User Navigation

Mr Darren Clowes[1&3], Mr Ray Dawson[1], Dr Steve Probets[2], Dr Julian Johnson[3], Mr Chris Holmes[3]

[1] Computer Science Department, Loughborough University, Leicestershire. LE11 3TU
[2] Information Science Department, Loughborough University, Leicestershire. LE11 3TU
[3] BAE Systems (Systems Engineering Innovation Centre), Leicestershire. LE11 3TU

d.clowes@lboro.ac.uk
r.j.dawson@lboro.ac.uk
s.g.probets@lboro.ac.uk
julian.johnson@baesystems.com
c.k.holmes@lboro.ac.uk

Abstract: Most of the knowledge that has been captured and stored for future use is held in traditional, text-based documents. Some of these documents can be very large, which can make the retrieving of the knowledge difficult and time consuming. In large technical documents such as military standards, navigation through these documents is usually by traditional linear scrolling or the use of a hierarchical menu. These documents commonly contain semantic knowledge within their text that leads to non-traditional navigation. This non-traditional navigation is normally achieved through users jumping between sections.

This pilot study has been conducted with BAE Systems. It uses the military standard known as the Tactical Data Link (TDL), Link 16 standard. This document is over 7000 pages, which highlights the potential problems that can be experienced in document navigation. Users of these large technical manuals are often searching for specific segments of knowledge from within the text. The text contains sections, which may or may not be amenable to semantic modelling. The semantic knowledge usually results in users having to cross-reference between several sections.

The document is currently available in Adobe PDF and Microsoft Word formats. These formats, are described in terms of their document navigation aids, as well as an experimental prototype HTML format developed at BAE System designed to display the document and allow for more advanced navigation strategies such as using the document semantics. The document structure model being used is discussed and an extracts of the supporting semantic model are discussed in relation to using the semantic knowledge of the document to aid user navigation. The 3 formats are then evaluated against each other, with examples highlighting the amount of navigation required to perform tasks.

This paper questions the use of the Microsoft Word format, and due to how Adobe PDF performs copy and paste believes the HTML prototype as the best solution to move forward with. It acknowledges shortcomings of the prototype HTML system with its lack of a page concept which is the focus of future work at BAE Systems. The semantic enhancement is seen as beneficial, enhancing the speed of which certain information can be located. In particular the enhancement to the breadcrumb trail by suggesting alternative paths that could have lead to the current location shows promise. Empirical tests of semantic navigation with engineers is proposed to determine that there is sufficient real gain. The pilot study does suggest that work on the longer term goal of combining semantic information with document structure in hybrid semantic-document models is still worth perusing.

Keywords: Navigation, Semantic Knowledge, User Interface

1. Introduction

This pilot study forms an initial part of an larger investigation into Hybrid Semantic-Documents at BAE Systems. The goal of the research is to examine the potential exploitation that can be achieved through the combination of Semantic information and Document structure for large technical documents.

In engineering domains, the documentation is normally highly structured. Engineers are required to use, and have a detailed knowledge of these documents to perform their tasks. In complex engineering domains, these documents normally consist of large sections of prose. It can therefore be hard to understand the precise objective of the section. These sections generally contain elements of rigour, which could be used semantically and more ambiguous prose, which helps the readers understanding. These documents are also often only available in the form in which they were written.

In most cases, the document producer has not considered how the end engineer is going to use the document. It is written with the primary focus of dictating the information. For small documents (a couple of hundred pages) this static representation does not prove much inconvenience. However, imagine the inconvenience of having a document that is several thousand pages in size and the engineer is trying to track details of the faults they are investigating. The engineer in this case is likely to be jumping around the document, constantly having to revisit the index to find the next expected section. Many of these documents are highly structured, which enables efficient indexing and searching, but what is often ignored is that because of the domain, the documents also contain significant elements of rigour which can be used to create a semantic representation of the document as well.

Our pilot study is currently focused on one particular area of Military Standards known as Tactical Data Links (TDL). Working with engineers from the Military Air Solutions Division, we are experimenting with one essential standard used by engineers TDL Link 16. This standard does contain large sections of prose with semantic content, suitable for our research. It is also believed that findings from the Hybrid Semantic-Documents research will be applicable to not only Military Standards but other technical areas, such as medical, legal and civil automotive/aeronautical.

1.1 Tactical Data Links (TDL)

The TDL domain is a collection of related technologies designed to work within the command, control, communications, computers, and intelligence (C4i) domain used in the dissemination of information within a battlespace to support joint and combined operations. Due to the of these operations several variations have been evolved to interface with specific unit types within the battlespace. As such, the TDL domain is often referred to as a family of standards. The DoD identifies these variations of the TDL (TADIL) domain via a postfix identifier (A, B, C, F, J, K, and M). These variations may differ in waveforms, bandwidths, protocols and capabilities. For the purposes of this pilot study Link 16 or TADIL-J standard will be used, this is described in MIL-STD-6016C.

The TDL standards are written to cover the full spectrum of applications. When a TDL is to be implemented on a platform (generically used to refer to the asset that the TDL is being integrated into i.e. ship or aircraft), only the required subset for the platform’s role is implemented. For example a transport aircraft does not need the fighter and bomber specific messages, so they are not implemented. Link 16 does not define platform roles and, as such, the subset selection is determined by the platform implementation team. Due to this, every platform is required to produce additional documents based on the standard.

The MIL-STD-6016C suffers from a number of shortcomings, the most significant being:

- It is a huge document (over 7300 pages).
- Data is duplicated across many sections and appendices.
- The sections and appendices are often interdependent
- Requirements are expressed in a mixture of natural language and semi-formal prose.
- Some of the notation leads to ambiguity and is potentially incomplete and inconsistent.

- It is not suitable for machine checking.

1.2 Aim

The aim of this pilot study is to investigate the idea of using the semantic knowledge from the textual prose to aid or provide new user navigation routes.

2. Related Work

Research conducted by Clowes et al. (2008) provides a recent review of literature addressing electronic representation and document navigation. Their review highlights the importance of spatial information in the reading of a document, with ability to generate a mental image of the document being critical. Piolat et al. (1997) found that having the page number visible also helped build this mental image. The review also discussed the issue of paging versus scrolling, with no pre-eminent technique. Dillion (1992) found only novices experienced difficulty using scrolling, where Pilot et al. (1997) noted paging as a better technique for generating a mental image. Clowes et al. (2008) also discussed the affect of disorientation in relation to a navigation aid. The main types of aids are termed as:-

- Hierarchical.
- Content lists, such as an alphabetical index.
- Linear, such as “next” and “back” links.
- Non-linear, such as linking to other sections that are not neighbouring.

Research reviewed finds that users get disoriented most when the navigation is focused on non-linear linking (McDonald and Stevenson, 1996, 1998, 1999). In fact, linear linking is considered most efficient for both performance of recall actions and limiting a user’s disorientation. Hierarchical and contents-list linking is considered to sit between the extremes of linear and non-linear linking. In addition, Park and Kim (2000) also found temporal aids such as a breadcrumb trail and history list coupled with spatial aids such as a hierarchical menu key helped to limit a user’s disorientation.

Vaughan and Dillon’s (2006) work in “genre” suggests it plays an important role. A site conforming to its accepted genre enabled users to recall the structure better, and this should aid the spatial location of text. It is also believed to allow users to generate a better mental image of the site. It is also important to note that the non-conforming site did improve over time, however, it was not able to match the performance of a genre conforming site.

Given that the TDL Standard is available as a PDF, the findings of Gulbrandsen et al. (2002) are interesting. Users appeared to find reading PDF documents more difficult than paper, whereas their HTML readers did not experience this difficultly. Unfortunately the study is unable make this conclusion as it primarily investigated the differences between paper and screen.

3. Current System

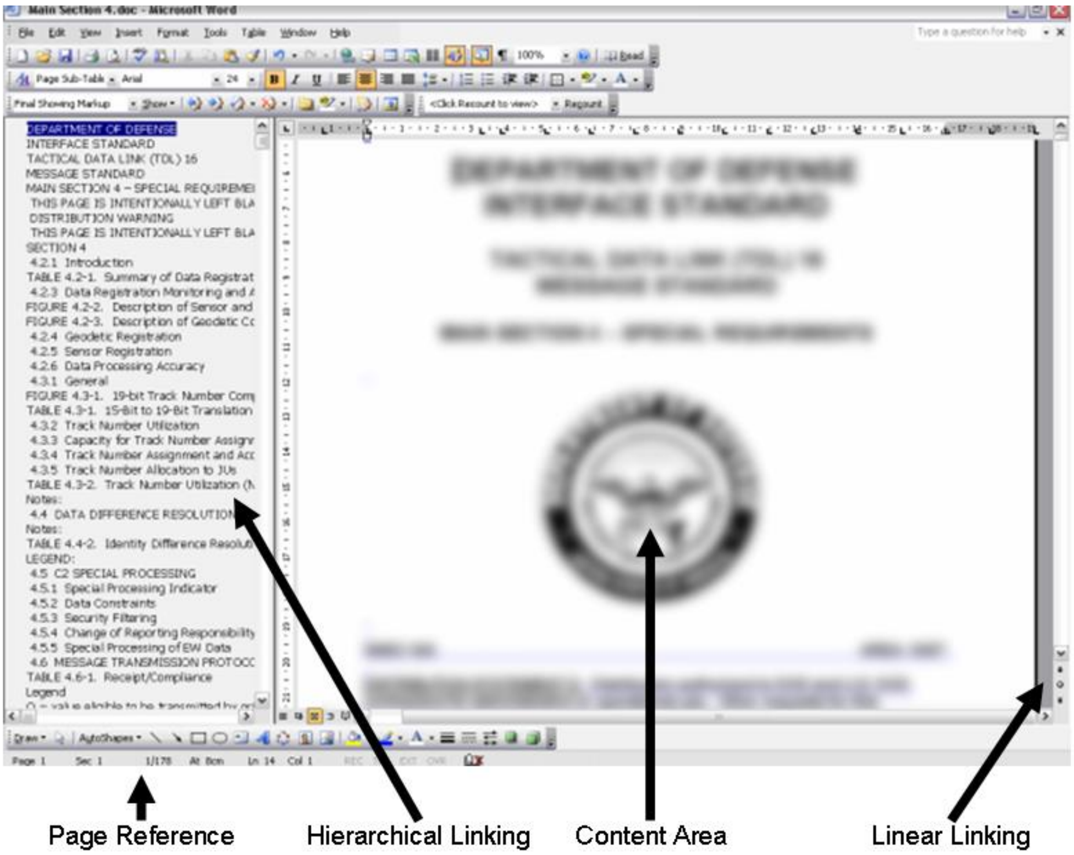
The MIL-STD-6016C is currently only available to engineers in electronic format. This is mainly due to the 7300 pages being equivalent to approximately 15 reams of paper, or enough to fill two entire filing cabinet drawers. Ventura (1988) argues that it is impractical to use paper based versions of large documents, given their weight and size. This is true for the MIL-STD-6016C, however engineer may choose to print out particular sections that they are working on.

In electronic format the DoD provides the standard in an Adobe PDF file or a collection of Microsoft Word files. It is important to note that the standard is not available in a single Microsoft Word document. The standard is spilt over 155 separate Word files which, when combined, represent the complete standard. As the standard defines all requirements for all roles of platforms, projects develop their own project version of the standard, which in most cases is just the platform required subset of the base standard. As such this project specific version is often maintained using only Microsoft Word. We are using the base standard MIL-STD-6016C as this does not restrict the work to one project. Before describing the feature of pilot application, we analyse the advantages and disadvantages of the MIL-STD-6016C in its current available formats.

3.1 Microsoft Word

Within Microsoft Word the navigation is primarily by scrolling. A Hierarchical menu can be generated through the document map function. However, the document map functionality requires the consistent use of Microsoft Word Styles such as heading 1 to generate the menu. The MIL-STD-6016C has no consistency in its textual formatting, in places it uses fixed width characters and spaces, where in others it uses formatting aids such as tables. Because the standard is spilt over several files the generated menu only works for that specific document.

Within the content area there are a number of non-linear links which jump to the relevant section in the relevant document, opening the file if needed. In creating the project specific document, the projects do only use one file, however they simply copy over the required sections of text from the MIL-STD-6016C. Microsoft Word also allows for page jumping through the “Go To” option available via the edit menu. When scrolling the page number is display during the process to indicate the position. The scroll to find a page does get increasing more difficult as the page count increases, in documents over 100 pages. Word does also support linear linking with its next and previous buttons. An illustration is provided in Figure 1 to show the main navigation aids and their approximate spatial location.



DEPARTMENT OF DEFENSE
INTERFACE STANDARD
TACTICAL DATA LINK (TDL) 16
MESSAGE STANDARD
MAIN SECTION 4 – SPECIAL REQUIREME
THIS PAGE IS INTENTIONALLY LEFT BLA
DISTRIBUTION WARNING
THIS PAGE IS INTENTIONALLY LEFT BLA
SECTION 4
4.2.1 Introduction
TABLE 4.2-1. Summary of Data Registrat
4.2.3 Data Registration Monitoring and A
FIGURE 4.2-2. Description of Sensor and
FIGURE 4.2-3. Description of Geodetic Co
4.2.4 Geodetic Registration
4.2.5 Sensor Registration
4.2.6 Data Processing Accuracy
4.3.1 General
FIGURE 4.3-1. 19-bit Track Number Com
TABLE 4.3-1. 15-Bit to 19-Bit Translation
4.3.2 Track Number Utilization
4.3.3 Capacity for Track Number Assign
4.3.4 Track Number Assignment and Acc
4.3.5 Track Number Allocation to JUs
TABLE 4.3-2. Track Number Utilization (N
Notes:
4.4 DATA DIFFERENCE RESOLUTION
Notes:
TABLE 4.4-2. Identity Difference Resolut
LEGEND:
4.5 C2 SPECIAL PROCESSING
4.5.1 Special Processing Indicator
4.5.2 Data Constraints
4.5.3 Security Filtering
4.5.4 Change of Reporting Responsibility
4.5.5 Special Processing of EW Data
4.6 MESSAGE TRANSMISSION PROTOCO
TABLE 4.6-1. Receipt/Compliance
Legend

(Picture distorted for security)

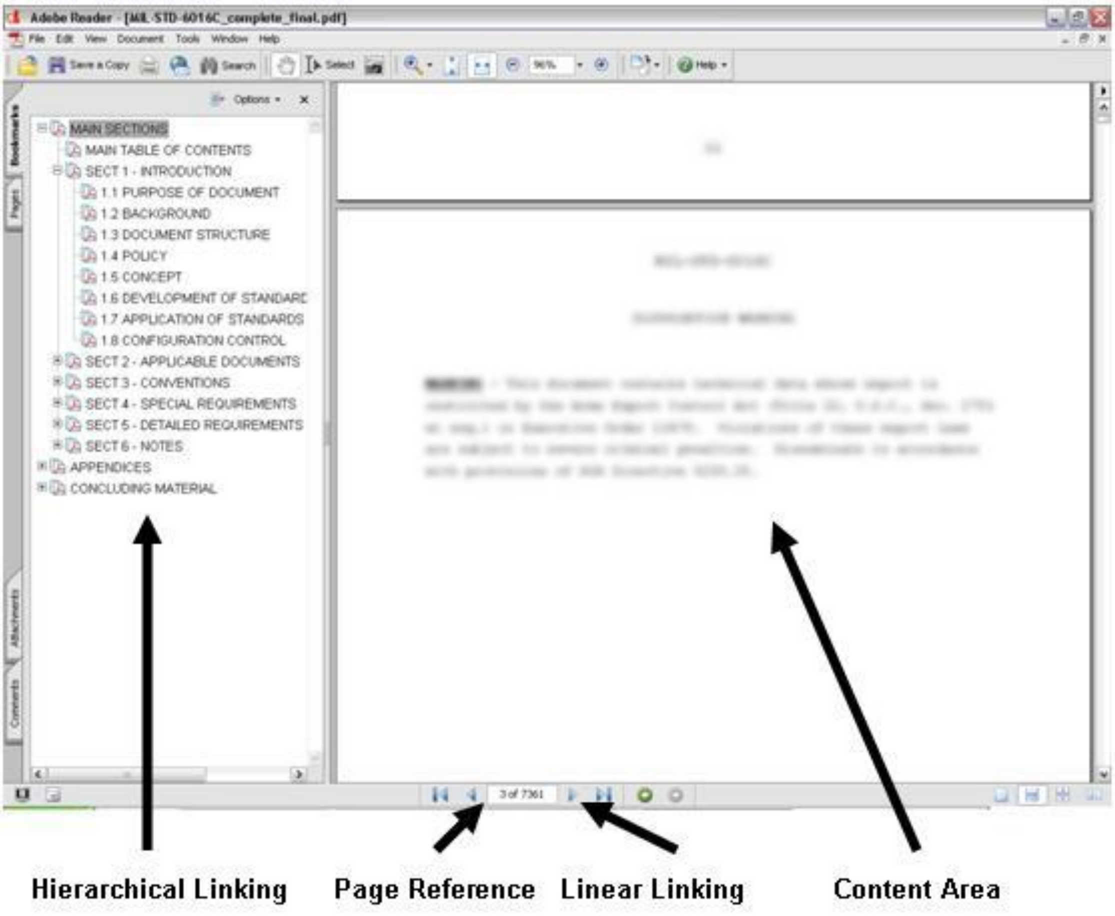
Figure 1: Microsoft Word Navigation aids

3.2 Adobe PDF

Adobe PDF is the other format that the TDL Link 16 standard is widely distributed in. Unlike Word, the whole standard is available in one file. Within this format it makes use of bookmarks to provide a hierarchical menu. The hierarchical menu has been constructed with three top level categories, “Main Sections”, “Appendices” and “Concluding Material”. The document creators have then limited the sub-level to a further two headings. This means the lowest level that can be navigated through this hierarchical menu is the section plus the main subsections (i.e. 4.2). The standard itself uses headings down to 4 levels (i.e. 4.2.1.1).

The Adobe PDF format makes use of both paging and scrolling with either technique being just as easy to use. Page jumping is easy, simply entering the required page number in the page reference information which is more emphasised than in Microsoft Word, which Piolat et al. (1997) suggest is should help limit disorientation.

The Adobe PDF format has limited non-linear linking. Again due how the document was created, in the contents list, the top level sections are linked to the position in the document, however, within the document the text is styled as a link but no functionality is behind the text, which is quite confusing for the user. A diagram is provided in Figure 2 to show the main navigation aids and their approximate spatial location.



(Picture distorted for security)

Figure 2: Adobe PDF Navigation aids

4. Experimental System

As a starting point for this project work at BAE Systems has focused on the development of a semantic model of the information and a document model of the structure. A document model has been developed, which allows the standard to be captured and stored in a relational database. This is then used to generate a HTML view of the standard via ASP.NET. The aim of the document model is to be able to represent the standard in a format familiar to the engineers through HTML (i.e. standard prose and tabular). Figure 3 shows the document model used. As can be seen from the figure, the model at the highest level consists of a set of main sections, which in turn consists of sections made up of block elements. The model allows for any object at these three levels to be referenced. Worth noting is that a table cell element is made of paragraphs which can be referenced, leaving the possibility to reference individual cells. In addition the model has no concept of a page.

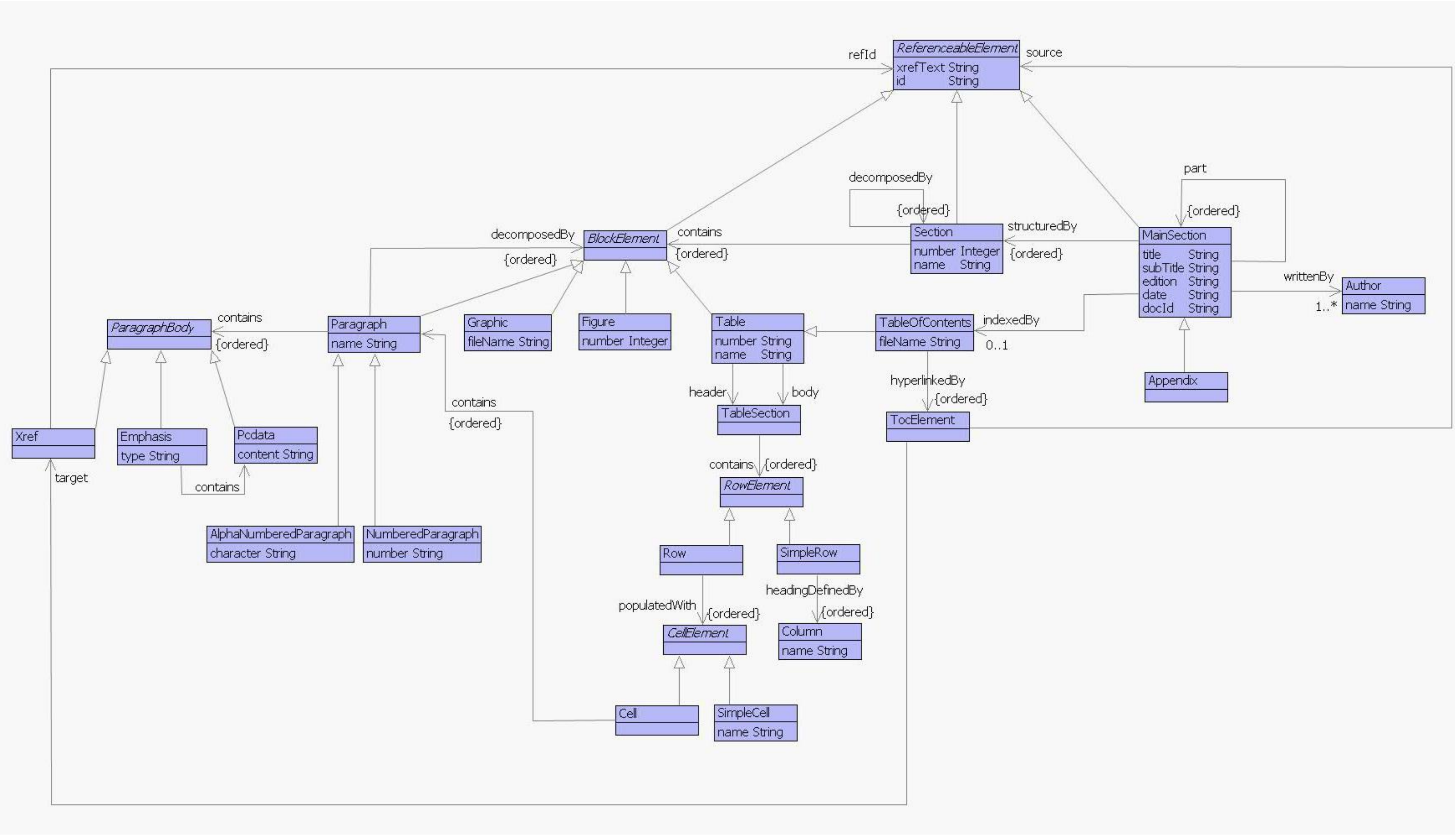
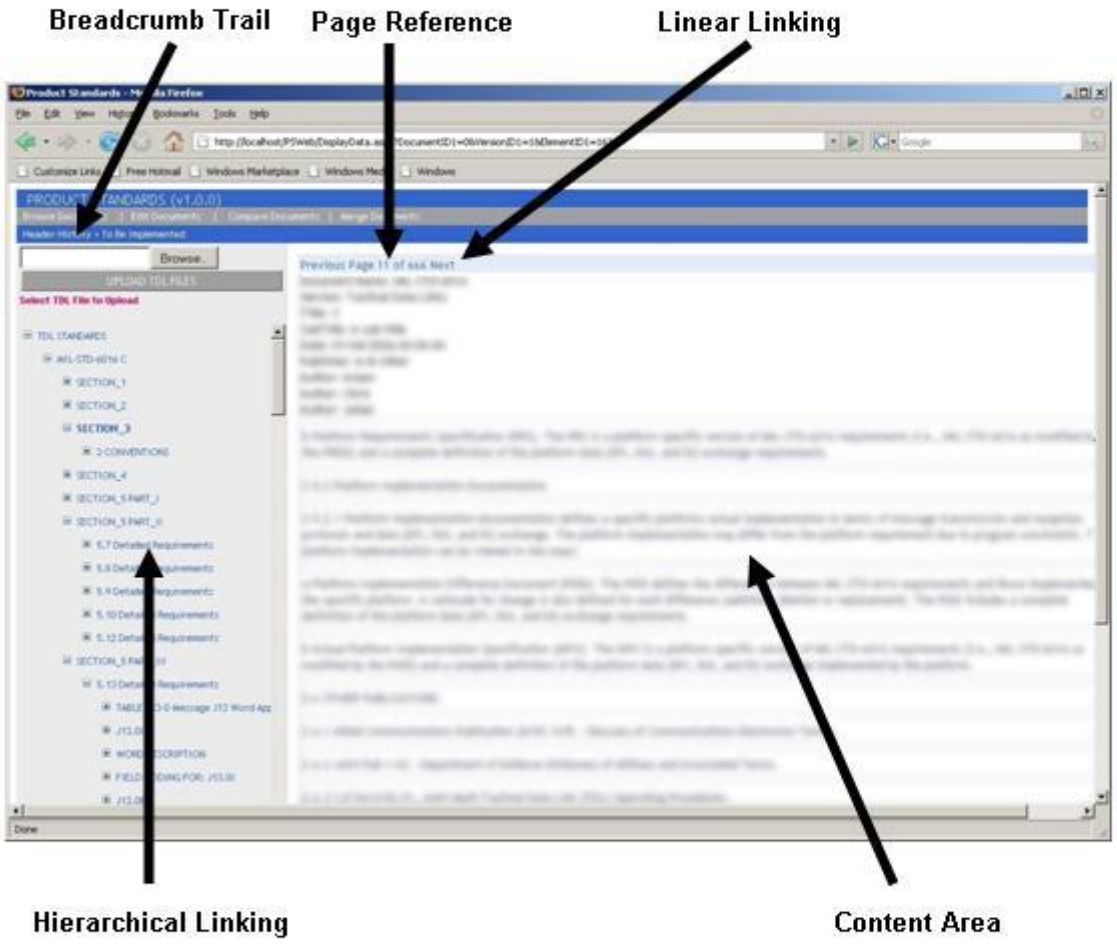

ReferenceableElement
xrefText String
id String
refId
source
decomposedBy
{ordered}
Section
number Integer
name String
structuredBy
{ordered}
part
{ordered}
MainSection
title String
subTitle String
edition String
date String
docId String
writtenBy
1..*
Author
name String
Appendix
decomposedBy
{ordered}
BlockElement
contains
{ordered}
Paragraph
name String
Graphic
fileName String
Figure
number Integer
Table
number String
name String
TableOfContents
fileName String
indexedBy
0..1
hyperlinkedBy
{ordered}
TocElement
ParagraphBody
contains
{ordered}
header
body
TableSection
contains
{ordered}
RowElement
Row
SimpleRow
headingDefinedBy
{ordered}
Column
name String
populatedWith
{ordered}
CellElement
Cell
SimpleCell
name String
contains
{ordered}
Xref
target
Emphasis
type String
Pcdata
content String
contains
AlphaNumberedParagraph
character String
NumberedParagraph
number String

Figure 3: BAE Systems document model

Using the model, an interface has been developed seen in Figure 4.This application makes use of a breadcrumb trail as suggested by Piolat et al. (1997) to aid a users awareness of their location within the document. The application does not allow for scrolling of the document, either the hierarchical menu and/or the linear links (previous and next page buttons) must be used. This is to limit the amount of data transmitted thus enabling acceptable page load times. As the model does not contain a concept for a page, a value of 15 block elements has been set as a page, resulting in a flexible page length, as not all block elements will use the same amount of space. The hierarchical menu is not constrained to a fixed depth, allowing users to drill down to the block element level provided they have a corresponding referenceable element within the model. Within the content area, non-linear linking around the document is functional.



(Picture distorted for security)

Figure 4: Experimental HTML Document view

A semantic view or semantic navigation is the planned extension of the prototype HTML application. Within the MIL-STD-6016C standard there is a wealth of semantic knowledge, ranging from the construction of messages, to their behaviour. A significant amount of work is being undertaken at BAE Systems in modelling this semantic knowledge. An example of the semantic information obtainable from the standard is shown in Figure 5. This is an extract, demonstrating the JMessage construct. At present, the latest work has been in developing the behavioural model as this area is the most promising for enhancing navigation. Within the behavioural area, are elements such as how sequences of JMessages combine to define different message uses and required transmission and receive orders. The development of the semantic model demonstrates some of the problems with standard such as the ambiguity of parts as some of the description can be open to user interpretation. It has also highlighted the amount of duplication. For example the dfi/dui definitions can be found within the description of a particular JWord when used by that JWord or listed in a data dictionary towards the end of the standard.

By using the semantic model it is believed that navigation aids can be developed that aid the user's ability to rapidly locate and move between elements in the document.

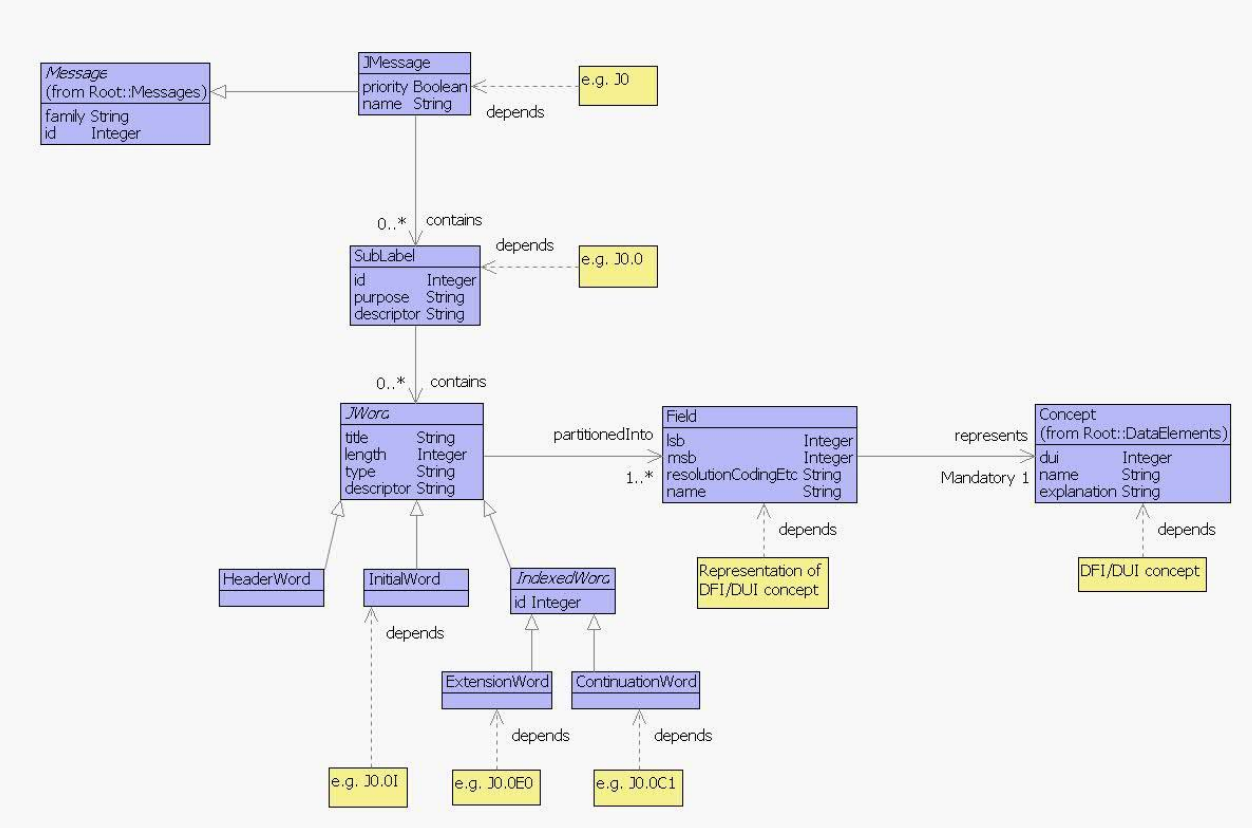


Figure 5: Message structure extract of the semantic model

The semantic models were analysed to determine what information could be obtained by traversing the model, such that the corresponding effort required in using the document versions could be compared against. An interface for the JMessage structure proposes using the use of drop down lists or listboxes. The interface has been proposed so that the user moves through a series of stages - after selecting an element, the next stage is populated with the relevant sub-level data. An example would be tracing a JMessage. In the first stage the user would select the JMessage e.g. J0, the next stage (using a drop down list or listbox) would then be populated with the SubLabels for J0. After selecting a SubLabel, e.g. J0.4, the next stage would list the available JWords (J0.4I, J0.4E, J0.4C1 & J0.4C2). After selecting the final stage the relevant information would be shown in the content area. A similar design concept for the behaviour is expected however given the complex nature of some of the transactions, this may prove unsuitable.

The temporal information within the HTML document view is used to display a “breadcrumb trail”. It is also proposed to enhance this information. The objective of the temporal information is to allow the user to recognise how they got to where they are and to retrace their steps. The enhancement proposed here is to leave the history button of a browser to provide the backwards navigation and the “breadcrumb trail” to show the most direct route to their location. Further to this, if it is possible to get to the current location by alternative means, then the possible parent nodes will also be displayed in the temporal linking area. This is expected to be most relevant with behavioural elements as a particular JWord could be part of multiple message uses.

5. Evaluation

The Microsoft Word version of the MIL-STD-6016C standard is possibly the worst format to use whilst reading or searching for information. Microsoft Word does have the functionality to produce a hierarchical menu through the document map function, but this is limited to the document itself and, again, having the standard split over several documents means the menu only displays the information in the current file. In addition the document map function appears to rely on consistent formatting using the in-built styles in Microsoft Word, which in the standard is not always guaranteed. On the positive side, Microsoft Word does have non-linear linking enabled within the document whereby the corresponding file to the link is opened on clicking, which the Adobe PDF file does not.

The Adobe PDF format has the significant advantage over Microsoft Word in that this version of the standard is in a singular file. This enables the Adobe PDF format to have a hierarchical menu covering the whole standard. However, this menu is limited only to three levels. The Adobe PDF is the best format for moving to specific already known pages (via page number).

The HTML version has the potential to use the benefits of both Microsoft Word and Adobe PDF whilst trying to limit their disadvantages. It does allow for a full depth hierarchical menu to be generated and used. The Standard is available in one file, however, due to the nature of web-based applications only a page is ever loaded to limit download times. This does mean that the HTML system does not allow for scrolling of the document like Adobe PDF and Microsoft Word. This is not necessarily detrimental as Clowes et al. (2008) did find that scrolling is not the best solution in all cases. The possible semantic improvements to the HTML have the greatest potential.

One of the main sections of semantic information is that of the JMessage structures. A JMessage within the system will be made up of a number of combinations of JWords with each JWord containing data elements. Semantically it is relatively easy to model this hierarchical structure of message detail. A sample of 89 messages from the standard was selected and using both the PDF and Microsoft Word versions of the document, they were navigated with two tests.

Test 1 aimed to determine the amount of navigation required simply to obtain a list of JWords for the given JMessage. The PDF file was easy to use, in that the hierarchical menu allows the user to navigate directly to the JMessage description with 5 steps. Only 18% of the JMessages required the user to navigate onto another page for the continuation of the JWord data. The Microsoft Word file was not easy to use. The user must navigate 10 pages to get to the index list entry for the correct section of the document. The index entry then loads another JWord file, from which an index link is followed to another part of the document which tells the user to find 3 further files which are not directly linked to from the document. There are therefore 3 index files which link to further files containing the information.

Test 2 aimed to discover the amount of further navigation required to determine the deeper detail in the JWords data element descriptions. This test required the same amount of navigation as the previous test plus an average of 33 linear page moves to generate a list of Data Element descriptions for a given JWord. The results are shown in Table 1

Table 1: Number pages to navigate

Page Moves	Number of Words
0-10	29
10-30	37
30-50	9
50-100	5
100+	9

The semantic model would allow for direct navigation through the JMessage structure bringing only a small decrease in the required navigation. By allowing the HTML systems hierarchical menu to traverse down lower than 3 levels, the document model version also allows for direct navigation of the JMessage structure due to the structure being spilt over subheading referencable by the hierarchical menu.

The behavioural semantics however is significantly more complex that a hierarchical structure. One of the simplest cases is that of pairing of tracks in the Weapons Coordination section. The pairing message is defined in J.10.6 which can be found on page 1838. To establish the behaviour of pairing messages the engineer must then analyse the transmit and receive tables. These are found starting at page 3422, the engineer must then move to page 3433 to establish that the Pairing is defined in Appendix K and has two transactions K.6.1 and K.6.2. The appendixes are located on page 5852. The appendixes define the constraints and stimulus for the transaction to occur. The actual values are located in the transmit and receive tables. The transmit data is defined on page 3642, and the receive on page 4154. The J12.0 or Mission Assignment is a lot more complex requiring even more movement and cross-referencing.

The behavioural model removes the need for a lot of the cross referencing and massive page jumping and the model can provide links to the relevant section of the document for example providing appendix K.6 with the J10.6 transmit and receive table as a section.

6. Conclusions

From analysing the current systems, it is questionable why the Microsoft Word version is so widely used and available, it has the significant disadvantages for navigation in that the standard is spilt across several files the support for a hierarchical menu is poor. The Adobe PDF version however, does not suffer from these; instead it appears to have some compile errors, with the limited menu depth and non-functional non-linear links. Surely it would appear that a sensible action would be to republish the standards formatted correctly and with deeper hierarchical menu structures in PDF. This would present a better solution for navigation. However, there are two reasons that this is not a valid path. The first problem with this approach is that the standard is used to create the platform specific documentation. When copying from an Adobe PDF file, the copied text does not preserve the paragraph formatting, instead it copies individual lines which must then be reformatted. Microsoft Word maintains this formatting which is why it is still widely used. The second reason is that the standard is maintained by the DoD. They are very unlikely to expend the resources on modify the links and republishing the standard. As such any re-issuing of the standard would need to be done at a local business level, which in turn would need to be performed for every release of an update to the standard.

The HTML system, is in away this local re-issuing. It has the disadvantages of the Adobe PDF document solved, the menu depth and non-linear links. It maintains formatting when being used to copy and the standard is available from one file. It is also more extensible than re-issuing a PDF document. The largest issue is with the lack of a page concept. The use of 15 block elements at first appears sufficient and acceptable for controlling the page. However, after a wider testing it was noted that for pages consisting of paragraphs it functioned acceptably, but in some circumstances such as in the “appendix B dfi definitions” where table concepts were high, the page was unexpectedly long.

As such work has started at BAE System looking at changing the document model to support a page concept and investigating if there is any significant need to consider the affect of a slightly different page construct to that regularly used by TDL engineers through the Adobe PDF format. For example by not have a paragraph spilt over a page. If engineers work simultaneously with both the Adobe PDF and this HTML standard could this slight change increase their disorientation?

The current work has looked at using the JMessage structural semantics and the message behaviour semantics to enhance navigation through the document. It has been found that little benefit is gained in using the JMessage semantics for navigating top-down views of JMessage Structures. However, there is the potential of the semantic model to easily allow bottom up navigation of JMessage structure. For example in fault tracking, the engineer may know the data element, from this they could then see what JWords and JMessages it could be from. The behavioural model is significantly more complex and requires much jumping around and cross referencing within the document. The potential for the semantics to be used in generating alternative views is significant here. The next stage is to conduct some empirical analysis with engineers to ascertain if it brings any real benefit.

This paper has highlighted the difficulties of extracting knowledge from very large, text based documents. Initial exploration of the methods of navigating one example of such a document, the MIL-STD-6016C, has yielded some insights into ways of aiding the user to find the knowledge and information required, but further research is needed to find the optimum way of navigating this and other large documents where important knowledge is held. This supports the long term goal at BAE Systems in researching hybrid semantic-document models, where the semantics and document structure can be used to present the most appropriate view of the information.

References

Clowes, D., Dawson, R., Probets, S. and Johnson J. (2008) “Quality Issues of User Interaction in Large Documents”, Paper at the Software Quality Management Conference 2008, University of Ulster, Northern Ireland, March.

Dillon, A. (1992) "Reading From Paper Versus Screens: A Critical Review of the Empirical Literature," *Ergonomics* Vol 35, pp. 1297 - 1326.

Gulbrandsen, P., Schroeder, T. and Nylenna, J. M. M. (2002) "Paper or Screen, Mother Tongue or English: Which is Better," *Journal of the American Medical Association* Vol 287 No. 21, pp. 2851-2853.

McDonald, S. and Stevenson, R. J. (1996) "Disorientation in Hypertext: The Effects of Three Text Structures on Navigation Performance," *Applied Ergonomics* Vol 27, No. 1, pp. 61-68.

McDonald, S. and Stevenson, R. J. (1998) "Navigation in Hyperspace: An Evaluation of the Effects of Navigational Tools and Subject Matter Expertise on Browsing and Information Retrieval in Hypertext," *Interacting with Computers* Vol 10 No. 2, pp. 129-142.

McDonald, S. and Stevenson, R. J. (1999) "Effects of Text Structure and Prior Knowledge of the Learner on Navigation in Hypertext," *Human Factors* Vol 40, pp. 18-27.

Park, J. and Kim, J. (2000) "Effects of Contextual Navigation Aids on Browsing Diverse Web Systems" 'CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems', ACM, New York, pp. 257-264.

Piolat, A., Roussey, J. and Thunin, O. (1997) "Effects of Screen Presentation on Text Reading and Revising," *International Journal on Human-Computer Studies* Vol 47, pp. 565-589.

Vaughan, M. W. and Dillon, A. (2006) "Why Structure and Genre Matter for Users of Digital Information: A Longitudinal Experiment with Readers of a Web-Based Newspaper," *International Journal of Human-Computer Studies* Vol 64, No. 6), pp. 502-526.

Ventura, C. (1988), “Why Switch From Paper to Electronic Manuals?”, *in* 'Proceedings of ACM Conference on Document Processing Systems 1988', ACM, , pp. 111-116.

Appendix D SQM 2009 Conference Paper

Paper published in 2009 in Software Quality in the 21st Century, pp. 195–206.

Quality Interaction with Database Driven Large Documents Over HTML

Mr Darren Clowes[1], Prof Ray Dawson[2], Dr Steve Probets[3], Dr Julian Johnson[4]

[1]BAE Systems/Loughborough University,
SEIC, Sir Denis Rooke Building,
Holywell Park,
Loughborough University,
LE11 3TU
d.clowes@lboro.ac.uk

[2]Computer Science Department,
Loughborough University,
LE11 3TU
r.j.dawson@lboro.ac.uk

[3]Information Science Department,
Loughborough University,
LE11 3TU
s.g.probets@lboro.ac.uk

[4]BAE Systems (Systems Engineering Innovation Centre),
SEIC, Sir Denis Rooke Building,
Holywell Park,
Loughborough University,
LE11 3TU
julian.johnson@baesystems.com

Abstract

At BAE System research is underway that requires the presentation of large complex documents using internet-based technologies. These documents can be thousands of pages in size. This paper reviews work that investigated whether a high quality interaction could be achieved by displaying the document as one scrollable HTML page. It found from the literature that a user will accept a 10-30 second load time for a web-page. The implementation was put through a series of optimisations to try and reach this target. It concluded that a paging system must be used for large documents.

1.0 INTRODUCTION

Increasingly, the design of complex engineered products and systems are becoming more reliant on computer-supported models/representations of information. Most of the knowledge that has been captured and stored for future use is held in traditional, text-based documents. Some of these documents can be very large, which can make retrieving the knowledge difficult and time consuming. In large technical documents such as military standards, considerable semantic knowledge is contained within their text. With documents like these making use of the semantic information as well as the document text can greatly improve an engineer's ability to locate, comprehend and utilise stored knowledge.

One such option for representing information to a large volume of users is via web-based techniques. For complex domains the amount of information can be extremely large, often amounting to thousands of pages. Migrating these traditional document-centric information sources to a web-based source poses several questions on how to achieve this. Web-based solutions are limited by the rapidly developing communication technologies that control bandwidth. This affects the speed at which information can be downloaded.

Technological advances mean modern day users are no longer willing to wait for information to load. The speed at which a page of information is displayed is now a serious deterministic factor for quality interaction. The introduction of web-based solutions has also brought with it new expectations for viewing and navigating pages of information.

This paper discusses the work undertaken at the Systems Engineering Innovation Centre by BAE Systems in assessing usability issues when displaying large complex documents over HTML. These documents are stored in a relational database according to a document model before being rendered using HTML.

2.0 BACKGROUND

At BAE Systems research is being conducted on the topic of Hybrid semantic-document models. This work is directed at pushing the boundary, feasibility and exploitation of a hybrid model approach to representing a complex system domain. A Hybrid model approach is being used to describe the combination of an underlying semantic model with mappings to a more traditional document model. This aims to act as a vehicle for deriving an unambiguous understanding of a complex domain that is currently represented in a conventional document-only based format.

A complex domain is defined as a domain where the documented materials are large in volume and contain rigorous elements that could be modelled semantically. Complex domains include Military and Civil Aviation, Military Standards, Automotive Standards, Legal Documents, Rail Timetabling etc. These differ from non-complex domains in which the documentation requires little or no rigour, or where the volume of information is so small that its representation will have negligible effect on engineers working practices. This project has focused on a Military Aviation Standard (Tactical Data Link 16) to prototype its theories.

Tactical Data Links (TDLs) are a family of standards. They document the communication standards for the sharing of battlespace information between allied assets. They are document in two variants NATO STANAGs and US Department of Defense MIL-STD documents. This project uses the MIL-STD for Link 16 (MIL-STD 6016C). The document for Link 16 alone is over 7,300 pages. The standard is available in a Microsoft Word Document or Adobe PDF format. It suffers from several shortcomings that a hybrid models approach could help solve. These include:

- Its sheer size
- Data is duplicated across many sections and appendices.
- The sections and appendices are often interdependent
- Requirements are expressed in a mixture of natural language and semi-formal prose.
- Some of the notation leads to ambiguity and is potentially incomplete and inconsistent.
- It is not suitable for machine checking.

As part of this project, a web-based tool is being developed to prototype exploitable features of the hybrid model approach. One of the first objectives for the project is the ability to render a version of the standard over HTML. This version will be stored via a Document Model in a database from which the HTML rendering will be obtained. The Document Model describes the structure of the document through elements such as sections, paragraphs, graphics etc. Later objectives will be to include the Semantic Model into the database and to achieve mapping between the data held in both models to drive new functionality and/or navigation abilities through the web-based application.

A high level of quality is needed between the interaction of the user and the information. Providing the information via HTML leads to the potential problems associated with limited bandwidth and the speed at which large amounts of data can be loaded.

3.0 LITERATURE

In assessing the quality of the interaction with web based documents, three key issues were identified from the literature:

- The time users will wait for a page to load.
- Whether the interface should page or scroll.
- Will loss of functionality affect the perceived interaction.

Before considering these issues an overview of developments in internet usage is provided as grounding for further discussion.

3.1 The Internet

The use of internet-based technologies such as HTML has exploded in the last decade. In 1995 there were approximately 16million users worldwide [1], with approximately 2600 domains [2]. In 2005 there were an estimated 1 billion worldwide users [1] and 350 million active domains [2]. In 2008 the number of internet users reached over 1.5 billion [1] with active domains reaching over 570 million [2].

Similarly the development of new technologies to support this exponential increase in users and domains has made rapid progress. The first commercial modem was released in 1962 and could achieve a speed of 300bps [3]. It was not until the 1990’s that the more familiar speeds of 14.4kbps-56kbps telephone line modems were developed, with 56kps arriving 1996 [3]. It took until the turn of the millennium for significant uptake of broadband services to be adopted in the UK. Broadband offered faster speeds in excess of 512kbps. Now some broadband suppliers are offering speeds of up to 50Mbps, a 100 fold increase in approximately 10 years [4].

Not only is the number of users and the speed of the underlying technology increasing but so too is the size of web pages[5]. King's study concluded that the average size of a web page has grown from 94k in 2003 to 312k in 2008, a 22 times increase on the average size in 1995. However, because the relative speed of internet connectivity has increased more than the relative increase in file size, King's belief that broadband users now expect faster response times seems sensible. King's study estimates that broadband users' average download time is now 2 seconds, down from 3 seconds in 2003.

3.2 Wait Times

The review of advances in the internet has shown that users now expect fast download speeds. To provide a high quality interaction a web-based solution must conform to these expectations. Neilson was one of the earliest researchers to investigate the effect of waiting times of page loads in the mid 90's [6,7]. Nah [8] provided a comprehensive review into tolerable waiting times. Her review found that users expect a computer response to occur within 0.1-10 seconds. However for web-pages this perspective has shifted; she found users were prepared to wait much longer, between 8-41 seconds. There is some supporting research that suggests that 30 seconds is upper limit [8,10,11]. However most other studies believe that a web user is willing to accept an 8-10 second delay in loading a page [7-9].

Nielsen listed download speed as one of the top ten problems of web usability [7]. In 2004[6] Neilson estimated that wait time would still remain a problem until 2011 because of technology, however in 2007 [12] waiting times no longer made his list of top ten problems for usability.

Most research suggests that having to wait is a negative; Slevidge, Chaparro and Bender [11] agree with this for the majority of cases. However, their research does lead them to suggest that waiting can be a positive, especially for sites which are cognitively demanding. In these cases users use this time to process the information and to plan their next moves. This would support a slightly longer time delay in loading information in complex domains.

In general the wait time does not affect a users opinion of the material, but it does increase the level of frustration. The level of frustration is likely to be affected by [13]:

1) The length of the wait
2) The level of uncertainty of the wait
3) The level of information provided on the wait
4) When the wait occurs (i.e. at the start or in the middle)

Several pieces of research believe that the key to limiting this is to provide information on the delay or a progress report [9,13]. Providing this information during the wait can reduce the level of frustration a user feels. However the length of time should still not increase significantly. There does not appear to be any new research conducted despite the recent advances in broadband. King [5] notices that the average download time has decreased, but are users now only willing to wait less than 10 seconds for a web page?

3.3 Scroll or Page?

Unlike traditional documents there is no real concept of a page when considering web-based information. A page to a web-based system is simply a file, that file does not have multiple pages; the file is the page, no matter how much space is required to view it. As such, paging is a relatively foreign concept in web-based systems. However research has been conducted on the use of electronic documents in which theory can be extended to apply to web pages.

Many electronic documents employ scrolling. Dillon [14] found that novice users found comprehension harder when scrolling compared to paging, but concluded that there was no difference for other users. Piolat et al. [15] found that scrolling restricted a user's ability to form a mental picture and that a paging technique was much better for locating recalled information. They also found that although paging required more movement and actions, its users did not read the document any more slowly

In contrast to those who concluded readers gained a better understanding by using a paging layout rather than scrolling, Gulbrandsen et al. [16] found that users appeared to find reading PDF documents more difficult than paper, whereas HTML readers did not experience this difficultly. This would suggest that users performed better with scrolling. Unfortunately the study was unable make this conclusion as it primarily investigated the differences between paper and screen reading in different languages. Nielsen [17] believes that the use of PDFs in a web-based solution is a crime against usability. In 2003, he listed several reasons which he still believes true in 2008. From the list it can be assumed that Nielsen favours scrolling as a technique for navigating a web page.

3.4 Losing Functionality

There appears to be no research conducted into the effect of removing functionality in a new version of a product. Therefore, should newly developed web-based applications mirror traditional MS Word or Adobe PDF versions? The review of navigation features in TDL document applications by Clowes et al.[18] showed that PDF and Word provide functionality to scroll the whole document or to move through a paging system as well as hyper-linking between elements of the document. An HTML system that displays the whole document will lose the concept of a page; however, graphics could be added to maintain the effect of a page. Similarly, if a system is unable to display the whole document then the scroll-ability is lost.

The closest area of related research is the effect of prior knowledge on using products. The use of prior knowledge on new product acceptance can be split into two groups [19]. These are Cognitive and Behavioural; both support different sides of the argument. Cognitive science believes prior knowledge is a positive benefit helping a user to use the product as tasks are performed automatically. In our case (developing a complex document browser), this may be a negative as the functionalities a user expects are no longer there. Behavioural science believes prior knowledge leads to poor judgement and over confidence of the users' ability to use the product. Interestingly, research [19] found that expert users were likely to adapt to new products if they were of the opinion that the product was new rather than updated. A technique proposed to aid users' transition is to provide clues to any new or heavily changed features.

Prior knowledge also affects a reader's preconception, resulting in an expected navigational structure. Most recently, Vaughan and Dillon's [20] work has termed this "genre" in an in-depth comparison of a genre conforming news website, against non-conforming versions. The results suggested that genre does play an important role. Prior knowledge is therefore limiting a users disorientation and ability to locate and recall information [20,21]. It is also believed that 'genre' allow users to generate a better mental representation of the site. As such if the users are experienced with a type of interface, then changing it will result in low acceptance.

4.0 AN INVESTIGATION INTO DOWNLOAD TIMES FOR TDL DOCUMENTATION

Having developed a web-based tool to maintain a hybrid model of a complex document, the challenge was to engineer a solution such that a high quality interaction can be achieved from the perspective of reducing waiting time to access information.

4.1 Web System

A document model was developed which modelled the structure of the document. This model was then mapped into tables of a relational database. The model consisted of elements such as section, paragraph, table and graphic. From this database a rendering of the document could be constructed by reading an "element" table which provides the index to the table in which the element data is stored (i.e. paragraph table, section table, etc.). The model and data was populated into Microsoft SQL Server. ASP.NET was then used to access the database and retrieve the data to display as a web-page.

The ASP.NET solution was developed using a multi-layer architecture consisting of Presentation, Business Logic, Data Access and Data Storage Layers. A full discussion of the architecture can be found in [22]. As an overview a brief description of the layers is provided. The Presentation layer consisted of the page formatting of the ASP/HTML output. The Business Logic controlled what information should be fetched (such as all the data or subsets to create a page). The Data Access layer simply called Stored Procedures stored in the database or Data Storage layer, converting the SQL result into a custom object that could be communicated between the other layers. Both the Business Logic and Data Access were developed as DLLs to enable their use with other applications.

4.2 Equipment

The machine used for the investigation had an Intel Core 2 Duo T7300 2GHz CPU with 3GB RAM. The Hard Disk Drive was a Toshiba MK1637GSX 2.5-inch 160GB running at 5400rpm.The operating system was Windows XP Service Pack 3 and was running Microsoft Internet Information Services (IIS) 5.1. The SQL Server was Microsoft SQL Server 2005 standard edition using shared memory. All unnecessary services and applications were shut down on the machine during the tests including virus scanners etc. The web browsers installed included Microsoft Internet Explorer 7 and Mozilla Firefox 3.

4.3 Target

The Adobe PDF version of the MIL-STD-6016C standard takes between 2-5seconds to load. This ignores any amount of time to download the file. The file itself is 15Mb. A perfect solution should have load times comparable to this. However, as literature suggests, users are prepared to wait longer for web page loading, as such the target set for a high quality interaction was 10 seconds. It was also decided that the web-based solution would be classed as acceptable should a load time of under 30 seconds be achieved, however in such a case, information must be provided to the user on the progress of the wait (this will be ignored for testing purposes).

To test the application, the web-page was loaded on the server using Internet Explorer 7. This negated any delay associated with network topology and traffic. Internet Explorer 7 was previously loaded so as to remove any load time associated with the start-up of the browser. The timer started when the link to the standard was clicked and stopped once the page finished loading as indicated by the status bar. The web page was compiled in release mode so debugging did not affect the timings. Also any graphics and styling were disabled to test just the time to load the data. The file size was also observed as this gave an indication of how well the optimisation was working and the amount of further effort required. Note that a high performance home page was expected to only be around 30-35Kb [23,24], with the average page being 312Kb.

4.4 Optimisation

The Web solution was timed after initial development. It took 1min 39 seconds to load the subset of the standard that was loaded into the database. The file itself was 10Mb in size. Code optimisation was therefore needed in an attempt to reach the target of less than 30 seconds. As the application was developed using ASP.NET, one of the first suggestions was to remove/disable the viewstate used by ASP.NET. The viewstate enables ASP.NET to maintain state information for web-pages; it grows with the number of elements on the page. This optimisation actually produced no noticeable change in file size or speed.

Literature suggests many other strategies for improving the performance of web-pages [23,24]. The most relevant were selected and implemented. They included:

- Removing white space and characters used to format a document for easier human reading.
- Using default attributes.
- Reducing the number of elements on the page by removing design elements such as divs and spans where possible.
- Using short one character names for attributes, classes, etc.
- Removing any un-required closing tags when using HTML.
- Using fixed table widths for complex tables.

In the initial development, each table cell had rowspan and colspan attributes declared even if they were 1 (the default if not declared). Names and classes were given long descriptors suitable to aid a human reader (i.e. “Num” rather than “n”). Table 1 provides the load times observed and the file sizes after several of the listed optimisations.

Table 1: Overview of Optimisation Results

	Load Time	**File Size**
Initial Development	1min 39s	10Mb
Remove viewstate	1min 39s	10Mb
Remove white space	1min 38s	9.62Mb
Remove rowspan=1 & colspan=1	1min 35s	9.27Mb
Reduce number of elements (removing divs, spans etc)	1min 30s	8.31Mb
Use 1 char attribute names	1min 30s	8.26Mb
No closing tags	1min 29s	8.19Mb

Although the document has many large complex tables, fixed width optimisations were not implemented primarily due to the data being dynamically driven from the database. As such the layout of each table would be difficult to ascertain. In addition optimisations at this point were not showing significant increases in performance.

Having only achieved a load time of 1min 29s after optimising the constructed HTML, it was clear that further HTML optimisation was going to be of little benefit as reducing the file size by 2MB had resulted in only a 10 second saving. One of the items which was believed to be slowing down the performance was the volume of database queries. In the initial implementation, a query had to be made to the element table to determine which four possible tables the data was found in. The implementation of the document model could however be changed such that the element table held all information across all four types of data (paragraph, table, section, graphic), however this would result in massive amounts of blank or null fields in the database. It was thus clear that displaying the whole document on a single HTML page would not enable a high quality interaction. Alternatively it was decided that a HTML paging solution is required which will reduce the amount of information per web-page. By reducing the volume of transmitted data the web-pages should load within acceptable time limits.

To segment a large document in pages break points need to be considered. Within large technical documents there are many logical break points – either at sections or sub-sections boundaries, or simply on a fixed or user-defined number of elements. It was observed from MIL-STD-6016C, that sections can be many 1000's of pages long, which leads to the same issues in trying to optimise the application for fast load times. As such a fixed number of elements per page was used. Testing with values of 10 or 20 elements results in page loads of less than 10 seconds. Problems did occur when treating a table as an element. Tables in technical documents can be extremely large and as such a page with several large tables took over 10 seconds to load.

5.0 EVALUATION

The testing environment was very artificial, in a real-world deployment, results would be expected to be even slower. However as the results did not approach the targeted load time of 30 seconds in this environment no testing in a realistic environment was required. Even though literature suggests that a load time of 30 seconds would be acceptable, this is questionable given the speed at which the technology underlying the internet is developing. Users' opinions may have changed in the last few years.

The optimisations focused solely on improving the constructed HTML, no consideration was given to optimisations of the database. The database was however configured to use Stored Procedures which should improve performance and the table design had a low volume of wasted space. To improve performance the design and construction of the database is perhaps of more significance than the HTML optimisation. The current design leads to many database calls to display just one element to the screen. The document model used should be reviewed, however, it is anticipated that no other suitable solution exists to deal with the exclusive-or relationship between an element and its type.

ASP.NET was been chosen as the technology because of its abilities to meet other requires outside of performance [22]. However development in another technology may improve the load times. Alternatives such as PHP [24] or AJAX [25] could be considered.

It was clear that for large documents, simple HTML optimisations can make a big difference. Removing the white space, using defaults and limiting characters saved approximately 2Mb of information with a 10 second saving. However, some HTML optimisations are not practical or suitable for database driven large documents such as MIL-STD-6016C. Fixing table widths are impractical as the required widths for columns are not static. This results in the HTML browsers having to process a table twice. Once to obtain the data and secondly to assign dimensions.

Even with database optimisations it is therefore believed that for large database driven documents, a single HTML page of the information is not a viable option. In this experiment only a subset of the 7300 pages was imported into the database. This subset still took over 1 minute to load. As such a paging system would seem to be required for such documents. Although this would result in the loss of functionality of the traditional use of the information (such as not being able to scroll through the document), it is likely that this will be less of an issue given the frustration that a user would experience in waiting for the document to load. Literature suggests that there is likely to be some loss of efficiency with any new interface, however with the possible development of advanced features through the use of hybrid models, it is hoped that these new feature will compensate for any loss.

As such in the long term it is hoped that a HTML-based paging interface with hybrid model improvements will become the ‘genre’ for interaction with large complex technical documents. Engineers’ currently experience some difficulties in using the PDF document. A system which provides less difficulties (such as easier navigation) will hopefully become the adopted ‘genre’ quicker.. Nielsen [17] also believes that PDF layout should only be used for print and that HTML should take its own form that supports the dimensions of a screen rather than paper. Therefore, a HTML interface that mirrors that of a PDF is not an ideal solution.

Given that a HTML-based paging approach seems preferable, the question arises as to how to split the information into pages. Within large technical documents there are many logical break points. It would seem most logical to break on sections as each page would then be a clearly segmented section. However, as can be observed from the MIL-STD-6016C, sections can be very large, which leads to the same issues in trying to optimise the application for fast load times. That led us to consider using sub-sections, but that led to the question - at which level do you break? Therefore, this project has moved in favour of a fixed number of elements - although problems have occurred when treating tables as a singular element. For example a page displaying 20 elements, of which 10 could be large tables, results in a very long page. Currently the team is looking at counting a row as an element and splitting tables if required. This is on-going work, and consideration needs to be given to the effect of splitting a table and the ease with which the information can be observed across the multiple pages it would span.

6.0 CONCLUSIONS

HTML is mark-up intensive resulting in bloated documents. For large documents, simple changes can make significant improvements. Examples such as reducing the number of characters used for names in attributes can save a large amount of space, thus saving time.

Nielsen [17] comments that PDF is designed for print reading and that this type of layout and interface makes it difficult for screen reading. Mirroring the interface and layout of a PDF document is therefore not necessarily a sensible solution. Although literature suggests loss of familiarity will reduce the effectiveness of an application, clear clues on how to overcome these losses should result in a negligible effect. Without a paging system for large complex documents, load times are likely to exceed the period of time a user is prepared to wait. Nielsen [26] also suggests that web-pages should be kept small with occasional large sections. This too would support the concept of paging large complex documents in providing a high quality interaction. Therefore the remaining work in this project will make use a paging solution with a fixed number of elements per page. The number of elements displayed will be customisable by the user to fit the screen restrictions.

7.0 REFERENCES

1. Miniwatts Marketing Group (2008), “Internet Growth Statistics” http://www.internetworldstats.com/emarketing.htm [accessed: 01/03/2009]
2. Internet Systems Consortium (2009), “The ISC Domain Survey”, https://www.isc.org/solutions/survey [accessed: 01/03/2009]
3. Dunn J. (1997) “Introduction and History of Modems” http://www.dementia.org/~julied/tele2100/intro.html [accessed: 01/03/2009]
4. cable.co.uk (2009) “Broadband History”, http://www.cable.co.uk/guides/broadband-history/ [accessed: 01/03/2009]
5. King A. (2008) “Average Web Page Size Triples Since 2002” http://www.websiteoptimization.com/speed/tweak/average-web-page/ [accessed: 01/03/2009]
6. Nielsen J. (2004) “The Need for Speed” Alertbox. http://www.useit.com/alertbox/9703a.html [accessed: 01/03/2009]
7. Nielsen J. (1996) “Original Top Ten Mistakes in Web Design” Alertbox. http://www.useit.com/alertbox/9605a.html [accessed: 01/03/2009]
8. Nah F.F. (2004) “A study on tolerable waiting time: how long are Web users willing to wait?” Behaviour & Information Technology,23(3) pp.153-163
9. Palmer J. (2002) “Designing for Web Site Usability”, IEEE Computer, 35(7) pp.102-103
10. Rose G.M., Meuter M.L. and Curran J.M. (2005), “On-line Waiting: The Role of Download Time and Other Important Predictors on Attitude towards E-retailers”, Psychology & Marketing, 22(2) pp.127-151
11. Selvidge, P.R., Chaparro, B.S. and Bender, G.T., (2002), “The World Wide Wait: Effects of Delays on User Performance”, International journal of Industrial Ergonomics, 29 pp.15-20
12. Nielsen J. (2007) “Top Ten Mistakes in Web Design” Alertbox. http://www.useit.com/alertbox/9605.html [accessed: 01/03/2009]
13. Dellaert B.G.C and Kahn B.E, (1999), “How Tolerable is Delay? Consumers’ Evaluations of Internet Web Sites After Waiting”, Journal of Interactive Marketing, 13(1), pp.41-54

14. Dillon, A. "Reading From Paper Versus Screens: A Critical Review of the Empirical Literature," Ergonomics (35), 1992, pp. 1297 – 1326

15. Piolat, A., Roussey, J. and Thunin, O., (1997), "Effects of Screen Presentation on Text Reading and Revising," International Journal on Human-Computer Studies (47), pp. 565-589.

16. Gulbrandsen, P., Schroeder, T. and Nylenna, J. M. M., (2002), "Paper or Screen, Mother Tongue or English: Which is Better," Journal of the American Medical Association 287(21), pp. 2851-2853.

17. Nielsen J. (2008) "PDF: Unfit for Human Consumption" Alertbox. http://www.useit.com/alertbox/20030714.html [accessed: 01/03/2009]

18. Clowes, D., Dawson, R.J., Probets, S., Johnson, J. and Holmes,C., (2008), "Pilot Studies in using the Semantic Knowledge of Information in Large Technical Documents to Aid User Navigation", Proceedings of 9th European Conference on Knowledge Management 2008, pp.129-138 Southampton UK.

19. Wood S.L. and Lynch, Jr. J.G. (2002), "Prior Knowledge and Complacency in New Product Learning", The Journal of Consumer Research, 29(3) pp.416-426

20. Vaughan, M. W. and Dillon, A., (2006), "Why Structure and Genre Matter for Users of Digital Information: A Longitudinal Experiment with Readers of a Web-Based Newspaper," International Journal of Human-Computer Studies 64(6), pp. 502-526.

21. Galletta, D.F., Henry, R.M., McCoy, S., and Polak, P., (2006), "When the Wait Isn't So Bad: The Interacting Effects of Website Deay, Familiarity, and Breath" Information Systems Research, 17(1), pp.20-37

22. Clowes D. et al, (2009), "A Multi-layered Approach to the Development of Custom Tactical Data Link Applications" Proceedings of International Conference of Systems Engineering Research, Loughborough UK (Unpublished)

23. Yahoo Inc, (2009), "Best Practices for Speeding Up Your Web Site", Yahoo Developer Network http://developer.yahoo.com/performance/rules.html [accessed: 01/03/2009]

24. King A, (2001), "Extreme HTML Optimization" WebReference.com http://www.webreference.com/authoring/languages/html/optimize/ [accessed: 01/03/2009]

25. Paulson, L.D., (2005), "Building Rich Web Applications with AJAX", IEEE Computer, 38(10), pp.14-17

26. Nielsen J. (2007) "Writing for the Web" Alertbox, http://www.useit.com/papers/webwriting/ [accessed: 01/03/2009]

Appendix E SQM 2008 CONFERENCE PAPER

Paper published in 2008 in Quality Issues for Business, pp. 37–48.

Quality Issues of User Interaction in Large Documents

Darren Clowes[1], Mr Ray Dawson[2], Dr Steve Probets[3], Dr Julian Johnson[4]

[1]BAE Systems/Loughborough University,
SEIC, Sir Denis Rooke Building,
Holywell Park,
Loughborough University,
LE11 3TU
d.clowes@lboro.ac.uk

[2]Computer Science Department,
Loughborough University,
LE11 3TU
r.j.dawson@lboro.ac.uk

[3]Information Science Department,
Loughborough University,
LE11 3TU
s.g.probets@lboro.ac.uk

[4]BAE Systems (Systems Engineering Innovation Centre),
SEIC, Sir Denis Rooke Building,
Holywell Park,
Loughborough University,
LE11 3TU
julian.johnson@baesystems.com

Abstract

Quality is important for any system, perhaps even more so for safety critical systems such as military aviation. A military standard used in aviation suffers several critical shortcomings with the key issue being the shear size (over 7300 pages). This results in difficulties for engineers in traversing and locating information, hindering quality. This paper presents a critical review of literature focused on issues and benefits of electronic representation of documents, the cognitive overhead of users and the disorientation of users during navigation. It forms part of a project by BAE Systems looking at techniques to develop advanced forms of navigation through such documents. The review reveals several unanswered questions. Two of the main questions being, given the exponential increase in internet usage do the issues raised still exist? Does the size of the document effect disorientation and cognitive overhead?

1.0 INTRODUCTION

Quality is important for any system, perhaps even more so for safety critical systems such as military aviation. An aspect in military aviation is the Command, Control, Communications, Computers, and Intelligence (C4I) system, and part of this is the Tactical Data Link (TDL) family of standards. These standards describe the communication process for passing information in relation to the battlespace. One such standard is Link 16 which is described in the Department of Defense’s MIL-STD-6016C. Within the defence sector there is a significant movement towards improving the quality via network enabled systems. The interoperability of these systems can only be achieved if the systems share a common understanding of the meaning of the standard. The MIL-STD-6016C suffers from a number of shortcomings making a common understanding difficult and hindering quality. The most significant being:-

- It is a huge document (over 7300 pages).
- Data is duplicated across many sections and appendices.
- Requirements are expressed in a mixture of natural language and semi-formal prose.
- Some of the notation leads to ambiguity and is potentially incomplete and inconsistent.
- It is not suitable for machine checking.

The critical issue is the shear size of the document. Developing quality products requires the engineers to be able to traverse the document easily to locate quickly and easily the information they are looking for. Currently the MIL-STD-6016C is only available in Microsoft Word and Adobe PDF file formats, with the exception of some custom applications developed by engineers for specific aspects of the standard.

Revising the document is not possible so engineers must work with the document in the development and maintenance of their products. The document is used in all stages of a product lifecycle, from requirements specification where a product will be contracted against a specific version and subset of the standard, to in-life maintenance. The implementation of Link 16 requires engineers to produce documents termed the Platform Requirements Specification and the Actual Platform Implementation Specification. These documents are primarily a reproduction of the standard, but including only the subset of material that is related to the product.

As part of a project by BAE Systems investigating novel document navigation, this paper provides a critical review analysing whether an electronic representation of documents is applicable and reviewing issues related to users cognitive overhead and disorientation. This forms part of a larger project at BAE Systems investigating the possible development of Hybrid Semantic-Document Models for modelling of document structure and the semantic knowledge within the text for advance forms of navigation.

This paper is organised as follows: Section 2 discusses the benefits and issues related to electronic representation of documents. Section 3 then reviews literature related to document navigation looking at both cognitive overhead and disorientation. Finally, section 4 concludes the review with a series of questions that the authors believe remain unanswered in the current literature available.

2.0 ELECTRONIC REPRESENTATION

The media used is believed to affect the comprehensibility of texts. It is commonly believed that reading from paper is both easier and more effective than reading from a computer screen. There has been much research in paper versus screen presentation of texts [1-6]. Most look at assessing the reasons behind the differences in comprehension with Dillon [2] giving a comprehensive critical review of the field. Ventura [1] however, considers the environmental and logistics of large technical manuals. She argues that it is impractical to use paper based versions of large documents, given their weight and size. This too is true for the TDL Link 16 with its 7300 pages being equivalent to approximately 15 reams of paper, which is enough to fill two entire filing cabinet drawers.

The comprehension studies are commonly based around Schumacher and Waller's [7] theory of outcome and process measures for usability. They spilt groups of measures into these two groups. Outcome measures are those based on the result of reading, such as speed and accuracy of recall. Process measures are those focused on the function of reading such as eye movement and navigation. Although many studies have been conducted, it is difficult to draw any solid conclusion from earlier literature as often the experiments suffered from poor experimental design and were of too narrow scope [2, 3]. The experiments often draw on only a small sample size between 10 and 25 people from which conclusions are drawn.

With the primary objective of this research being the improvement in usability of the TDL standard, outcome measures for speed, accuracy and comprehension are of most interest, with process measures such as eye movement [2, 7] being beyond scope. However, the navigational process issue and spatial considerations are of possible critical importance given the size of the document.

Previous research by Muter et al [8] had found that screen reading was 28.5% slower than reading from paper as too, did much of the literature reviewed by Dillon [2]. However, Muter and Maurutto [4] revisited this research to conclude that, given advances in the technology of screen quality, reading from a screen was comparable to reading from paper for both speed and comprehension. Although Dillon [2] draws the conclusion that reading from a screen is slower, his conclusion is based on older material while Muter and Maurutto come to a better, more up to date conclusion that, given the improved screen quality, reading from screen can be comparable to that of reading from paper

Annotation or note-taking while reading is believed to be a major aid in comprehension [3, 9]. This is primarily a benefit to those aiming to learn from the document, in particular the O'Hara and Sellen [3] experiment asked readers to write a summary after reading, which led the reader into taking notes. Although users of the electronic text were allowed to take notes in another document, users felt uncomfortable and dissatisfied. Given the advance nature of the TDL environment and shear size of the document this is not considered a contributing factor for comprehension in this domain. However, research has been conducted on developing systems that would allow the user to make freehand annotations to electronic documents as they read [9-12]).

The spatial layout of text has an effect on the process of reading, including the spatial layout in the environment and the spatial layout of the text on a page. The spatial layout assists readers to recall or search for information. Readers often create a mental picture of the document such that the location of information can be recalled in relation to their mental picture [2, 3, 5, 13, 14]. O'Hara and Sellen [3] found that readers used their sense of touch to acquire a sense of position within the document which is used to make the mental image. Piolat et al. [5] also found that having the page number visible also helped build this positional opinion. Both Dillon [2] and O'Hara and Sellen [3], found that this positional information was harder for electronic versions as a full page view is rarely achieved given screen sizes. Dillon does, however, record that performance can be improved through larger screens.

Because of this space limitation, most electronic documents employ scrolling. Dillon [2] found that novice users found comprehension harder when scrolling compared to paging, but concluded no difference for others. Piolat et al. [5] found that scrolling restricted a user's ability to form a mental picture and a paging technique was much better for locating recalled information. They also found that although paging required more movement and actions, its users did not read the document any more slowly. O'Hara and Sellen [3] discovered this movement can actually aid reading, the ability to quickly move to a cross reference page helps. This finding supported Dillon's [2] opinions that a key benefit to electronic texts is the possible use of multiple windows, which a user can rapidly switch between.

In contrast to those who conclude readers gained a better understanding by using a paging layout rather than scrolling, Gulbrandsen et al. [6] found that users appeared to find reading PDF documents more difficult than paper, whereas their HTML readers did not experience this difficultly. This would suggest that users performed better with scrolling. Unfortunately the study is unable make this conclusion as it primarily investigated the differences between paper and screen reading in different languages. It was only the Danish experiment that used PDF, the Norwegian and Swedish experiments used HTML. Similarly the screen size was not kept consistent so this too could have had an effect. This highlights Dillon's [2] and O'Hara and Sellen's [3] comments that experiments lack the necessary control over experimental design parameters. There appears to be little similar research, suggesting that particular applications can affect the screen reading ability.

Document navigation is possibly the most important factor for this project given the motivation of being able to provide advanced navigation routes through the semantic model of the document's detail. Determining the relative advantages of paper or electronic documents is not considered here, but a more in depth review of navigation research is provided in the next section.

3.0 DOCUMENT NAVIGATION CONSIDERATIONS

This project aims to develop advanced navigation opportunities by utilising the semantics of the information within the document. As such, the wealth of research on navigation is of critical importance. It can be segregated into two main categories, cognitive overhead and disorientation. Both topics are closely related in that the navigation structures affect the cognitive load and cognitive style will affect the degree of disorientation.

Studies show that the cognitive style of a reader affects their navigation [15-20]. They focus on the perspective of the reader as a learner and spilt users into field dependent and field independent learners. They can be characterised as in Table 1.

Field Independent		Field Dependent	
Characteristic	Preference	Characteristic	Preference

Active Approach	Prefer to use index to locate specific items	Passive Approach	Rely on map to impose structure
Analytical Tendency	Prefer depth-first paths	Global Tendency	Prefer breadth-first paths
Internally Directed	Prefer non-linear and flexible navigation	Externally Directed	Prefer linear and restricted navigation

Table 1: Field Dependent and Independent Characteristics
(Source: [15])

The common finding across all the studies is that the navigational process should be tailored to the reader’s cognitive style [16,17]. Independent learners are believed to favour a non-linear navigation with the freedom to explore on their own. In contrast, dependent learners require a linear structure relying on the designers to propose the intended path. There is little published evidence on whether these cognitive styles continue to be effective in other document uses, such as reference.

Interface tailoring on cognitive style has lead to the development of Adaptive Hypermedia [15-18]. This is where the interface is customised for the user given three main factors:-

Directed Guidance, where the link order is modified. For example, Field Dependents prefer a breadth first approach whereas independents prefer a depth first.

Link Disabling, as Dependent learners are easily disoriented and distracted by non-linear links, irrelevant links are either enabled or disabled.

Layout, where the structure is varied, i.e. a hierarchical map for dependents.

An interesting suggestion is the use of annotated links [16] such that the user can gain more understanding of the link before navigating away. However, Mitchell et al. [15] found no gain for tailored approaches compared to a well designed interface for all users, but still 75% of both dependent and independent learners preferred having a selection of navigation tools available.

Similarly to cognition, the domain context can affect navigation [15, 21, 22]. Prior knowledge affects a reader’s preconception, resulting in an expected navigation structure. Most recently, Dillon’s [21] work has termed this “genre” in an in-depth comparison of a genre conforming news website, against non-conforming versions. The results suggested genre does play an important role. The genre conforming version enabled users to recall the structure of the site better, and this should aid the spatial location of text. It is also believed to allow users to generate a better mental representation of the site. It is also important to note that the non-conforming site did improve over time, however, it was not able to match the performance of a genre conforming site.

Jakob Nielsen [22] believed that disorientation is a major problem for usability of hypertext documents. In general, the problem with disorientation is the users ability to get lost [24, 25]. It is generally considered that the problems revolve around users knowing where to go next, where they are, and how they got there. Park and Kim [19] put these factors down to two groups of contextual information, temporal and spatial. They suggest that the navigation system should aid users in re-establishing their sense of location through the spatial and temporal context. Hierarchical linking appears to be the suggested form of spatial information with a breadcrumb trail acting as a temporal provider.

Disorientation is mostly related to the type of navigation link aid used. The main types of aids are termed:

- hierarchical,
- content lists such as an alphabetical index,
- linear such as next and back,
- non-linear such as linking to other sections that are not neighbouring.

Research finds that users get disoriented most when the navigation is focused on non-linear linking [26-29]. In fact, linear linking is considered most efficient for both performance of recall actions and limiting a users disorientation. Hierarchical and contents list linking is considered to sit between the extremes of linear and non-linear linking. Most experiments seem artificial as the system only uses one navigation aid which is not common in reality. They are able to conclude some design considerations that limit disorientation, such as the use of page numbers and chapter headings. They are found to help a user in their recall of important information and aid the perception of size of the document.

An unusual navigation aid is the use of spatial maps [20, 23, 28]. They do not appear to be utilised widely in the World Wide Web, despite being considered good at lowering disorientation. McDonald and Stevenson’s [28] study interestingly found that this technique removed any benefit a user may have had from prior knowledge. Novice and expert domain users both achieved very similar levels of accuracy in recall and speed of navigation. It is not clear whether maps would be suitable for large documents as they could become cluttered, all currently reported research has been conducted with a relatively similar map with a low number of nodes.

Other considerations for navigation include the technique for moving between pages. Research suggests that variable zooming in and out dependent on the speed of scroll best mimics the scanning technique employed when reading from paper [30-32].

Weinreich et al [33] carried out a long term study of 105 days. Their view of navigation considered the web rather than a document and analysed the statistics of clicks such as back, links and submit clicks. Most interestingly for this study were their within-page navigation findings. They found 0.4% users would horizontal scroll to select a link. Key to successful rapid navigation was the ability of the user to quickly scan the information with limited scrolling. For this purpose they produced a screen pixel real estate chart (Figure 1). They believe that design should now cater for 1024x768 pixel resolutions and make the most use of high value pixels for navigation. Other design factors were found by Nielsen [23]. He found that subtle changes in background colour, to differentiate sections are not noticed, as a screen full of text does not allow a user to gain the spatial information required to recall the position of this text.

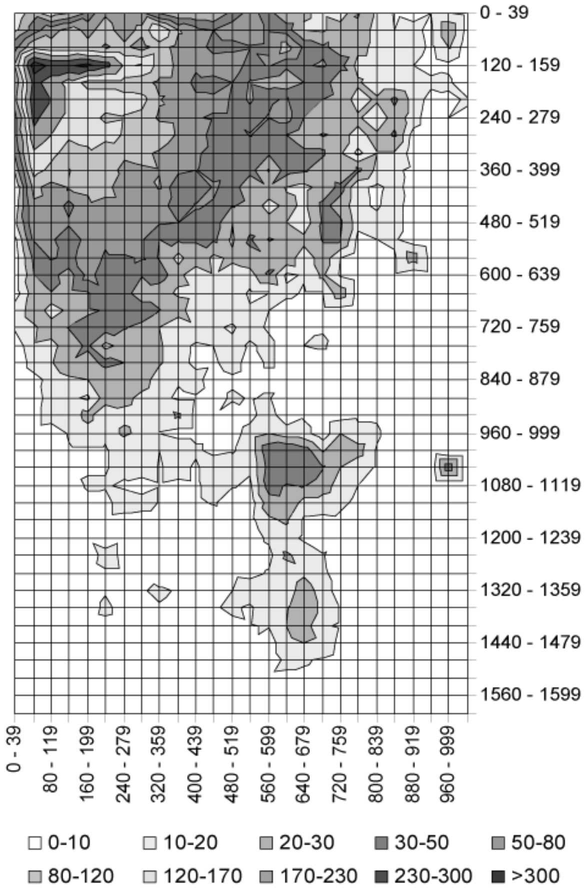


Figure 1: Pixel Real Estate Value
(Source: [33])

4.0 CONCLUSIONS

Electronic versions of documents are no longer limited by poor resolution and screen restrictions. Although paper has benefits for the reader, for larger documents an electronic version has outweighing benefits, such as search functions and portability. There are two main areas of research in document navigation, cognition and disorientation. Research in cognition has identified user attributes, indicating different reader capability, defined as Field Dependent and Field Independent. Study suggests that an interface should be well designed for all users or be customisable to suit the cognitive style. Disorientation is mainly influenced by the navigational aid being used, linear navigation is found to achieve the best performance with non-linear linking the worst. Studies also suggest that prior knowledge of the domain aids a user in their navigation, although the use of navigation maps helps novice users achieve similar results to expert users. However, some key questions remain unanswered from current literature:

1 The first question is, do the conclusions of all this literature still apply? The problem with hypertext is that the internet is still growing and growing at a remarkable speed. In fact, in 1995 it was estimated that there were 16 million users of the internet rising to 361 million in 2000. That is over a 2000% increase, or 450% increase a year if we compare with the December 2007 figures of 1.3 billion. The primary target user population for this project are likely to be Europe and North America who, when combined, account for 44%

or 572 million users. The penetration rate shows similar rapid advancement. This is defined as the percentage of the total population that has the basic knowledge of and available access to the internet. In 2007, North American had 71% and Europe 43% penetration. Within Europe, if only the European Union is considered, then this rises to 55% and, if only the UK, then it is 66% [34]. Again if we compare the UK penetration rate in 2000 (30%) we can see that this has almost doubled in seven years. All this suggests that internet usage is now more widely accepted, so possibly the elements of disorientation may no longer apply given the wider and more common place usage of the internet will enable readers to become used to the non-linear linking style of navigation.

2 Most literature, when investigating navigation, is focused on the user as a learner. Although some studies accept prior knowledge of the domain aids navigation, they still investigate the process as a learning activity. For the TDL domain the majority of use is for reference or a goal-directed information seeking context, whereby the user already knows a great deal and needs only to locate specific bits of information. This could be classified as the recall action, but in reality the user is not going to read all 7300 pages. It is unclear whether this change in intended goal will produce different findings to that of the research reported in the current literature. There is a limited investigation into goal-directed information seeking, such as by Gwizdka and Spence [13], but no author address disorientation and few address navigation.

3 Understandably, the published research investigates the hypotheses on small documents, usually around 4000-5000 words. No research has attempted to investigate whether there is a scaling effect in relation to document size. It is assumed that different navigation techniques will have different effects given the varying size of document. Furthermore, no reported research has investigated if the disorientation problem is multiplied with larger documents.

This paper has identified three significant unanswered questions regarding the navigation of very large text-based documents. Given that there is normally a considerable volume of knowledge and experience put into the development of such documents it is clearly beneficial for readers to be able to extract that information efficiently and effectively. This paper has shown that there is still important research to be carried out to give the best quality navigation and retrieval of information from these very large documents.

5.0 REFERENCES

1 Ventura, C. "Why Switch From Paper to Electronic Manuals?" 'Proceedings of ACM Conference on Document Processing Systems 1988', ACM, 1988, pp. 111-116.

2 Dillon, A. "Reading From Paper Versus Screens: A Critical Review of the Empirical Literature," Ergonomics (35), 1992, pp. 1297 - 1326.

3 O'Hara, K. and Sellen, A. "A Comparison of Reading Paper and On-Line Documents" 'CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems', ACM, New York, NY, USA, 1997, pp. 335-342.

4 Muter, P. and Maurutto, P. "Reading and Skimming from Computer Screens and Books: The Paperless Office Revisited?," Behaviour & Information Technology (10:4), 1991, pp. 257-266.

5 Piolat, A., Roussey, J. and Thunin, O. "Effects of Screen Presentation on Text Reading and Revising," International Journal on Human-Computer Studies (47), 1997, pp. 565-589.

6 Gulbrandsen, P., Schroeder, T. and Nylenna, J. M. M. "Paper or Screen, Mother Tongue or English: Which is Better," Journal of the American Medical Association (287:21), 2002, pp. 2851-2853.

7 Schumacher, G. and Waller, R. "Designing Usable Texts", in Duffy, T. and Waller, R., ed., Academic Press, Inc., 1985, pp. 377-403.

8 Muter, P., Latremouille, S., Treurniet, W. and Beam, P. "Extended Reading of Continuous Text on Television Screens," Human Factors (24:5), 1982, pp. 501-508.

9 Guimbretière, F. "Paper Augmented Digital Documents" 'Proceedings of ACM symposium on User interface software and technology', ACM, 2003, pp. 51 - 60.

10 Schilit, B. N., Golovchinsky, G. and Price, M. N. "Beyond Paper: Supporting Active Reading with Free Form Digital Ink Annotations" 'CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems', ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1998, pp. 249-256.

11 Price, M. N., Golovchinsky, G. and Schilit, B. N. "Linking by Inking: Trailblazing in a Paper-Like Hypertext" 'Proceedings of UK Conference on Hypertext', ACM, 1998, pp. 30-39.

12 Wellner, P. "Interacting with Paper on the DigitalDesk," Communications of the ACM (36:7), 1993, pp. 86-97.

13 Gwizdka, J. and Spence, I. "Indirect Assessment of Web Navigation Success" 'CHI '05: CHI '05 extended abstracts on Human factors in computing systems', ACM, New York, NY, USA, 2005, pp. 1427-1430.

14 Ahmed, I. and Blustein, J. "Navigation in Information Space: How Does Spatial Ability Play a Part?" 'IADIS International Conference on Web Based Communities', IADIS, 2005, pp. 119-125.

15 Mitchell, T. J. F., Chen, S. Y. and Macredie, R. D. "Adapting Hypermedia to Cognitive Styles: Is it necessary?" 'Proceedings of Workshop on Individual Differences in Adaptive Hypermedia', Published online by Department of Computer Science and Information Systems Birkbeck University of London, 2004, pp. 70-83.

16 Chen, S. "A Cognitive Model for Non-Linear Learning in Hypermedia Programmes," British Journal of Educational Technology (33:2), 2002, pp. 449-460.

17 Chen, S. Y. and Macredie, R. D. "Cognitive Styles and Hypermedia Navigation: Development of a Learning Model," Journal of the American Society for Information Science Technology. (53:1), 2002, pp. 3-15.

18 Dufresne, A. and Turcotte, S. "Artificial Intelligence in Education: Knowledge and Media in Learning Systems", in du Boulay and Mizoguchi, ed.,'World Conference on Artificial Intelligence and Education', IOS Press, 1997, pp. 287-293.

19 Park, J. and Kim, J. "Effects of Contextual Navigation Aids on Browsing Diverse Web Systems" 'CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems', ACM, New York, NY, USA, 2000, pp. 257-264.

20 Calcaterra, A., Antonietti, A. and Underwood, J. "Cognitive Style, Hypermedia Navigation and Learning," Computers & Education (44), 2005, pp. 441-457.

21 Vaughan, M. W. and Dillon, A. "Why Structure and Genre Matter for Users of Digital Information: A Longitudinal Experiment with Readers of a Web-Based Newspaper," International Journal of Human-Computer Studies (64:6), 2006, pp. 502-526.

22 Rouet, J. and Bigot, L. L. "Effects of Academic Training on Metatextual Knowledge and Hypertext Navigation," Metacognition and Learning (2), 2007, pp. 157-168.

23 Nielsen, J. "The Art of Navigating Through Hypertext," Communications of the ACM (33:3), 1990, pp. 296-310.

24 Elm, W. and Woods, D. "Getting Lost: A Case Study in Interface Design", Human Factors Society 29th Annual Meeting 'Human Factors Society 29th Annual Meeting', Human Factors Society, 1985, pp. 927-931.

25 Marchionini, G. "Hypermedia and Learning: Freedom and Chaos," Educational Technology (28), 1988, pp. 8-12.

26 McKnight, C., Dillon, A. and Richardson, J. "Hypertext: state of the art", in McAleese, R. and Green, C., ed., Intellect, Oxford, 1990, pp. 10-19.

27 McDonald, S. and Stevenson, R. J. "Disorientation in Hypertext: The Effects of Three Text Structures on Navigation Performance," Applied Ergonomics (27:1), 1996, pp. 61-68.

28 McDonald, S. and Stevenson, R. J. "Navigation in Hyperspace: An Evaluation of the Effects of Navigational Tools and Subject Matter Expertise on Browsing and Information Retrieval in Hypertext," Interacting with Computers (10:2), 1998, pp. 129-142.

29 McDonald, S. and Stevenson, R. J. "Effects of Text Structure and Prior Knowledge of the Learner on Navigation in Hypertext," Human Factors (40), 1999, pp. 18-27.

30 Sun, L. and Guimbretiиre, F. "Flipper: A New Method of Digital Document Navigation" 'CHI '05: CHI '05 extended abstracts on Human factors in computing systems', ACM, New York, NY, USA, 2005, pp. 2001-2004.

31 Jones, S., Jones, M., Marsden, G., Patel, D. and Cockburn, A. "An Evaluation of Integrated Zooming and Scrolling on Small Screens," International Journal Human-Computer Studies (63:3), 2005, pp. 271-303.

32 Cockburn, A., Gutwin, C. and Alexander, J. "Faster Document Navigation With Space-Filling Thumbnails" 'CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems', ACM, New York, NY, USA, 2006, pp. 1-10.

33 Weinreich, H., Obendorf, H., Herder, E. and Mayer, M. "Off the Beaten Tracks: Exploring Three Aspects of Web Navigation" 'WWW '06: Proceedings of the 15th international conference on World Wide Web', ACM, New York, NY, USA, 2006, pp. 133-142.

34 Internetworldstats "Internet Usage World Stats", internetworldstats, This is an electronic document. <http://www.internetworldstats.com/>. Date retrieved: February 4, 2008.