Contextual Annotation Framework using
Ontology-Based Contextual Annotation Service

By

Thanyalak Maneewatthana

A thesis submitted for the degree of Doctor of Philosophy

School of Electronics and Computer Science

University of Southampton

United Kingdom

September 2007

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS

SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

Contextual Annotation Framework using Ontology-Based Contextual Annotation Service

Thanyalak Maneewatthana

Current systems do not provide a context when sharing and reusing annotations. This research investigated how this may be achieved. This thesis provides an overview of the work that has been undertaken in the fields of Semantic Web, adaptive hypermedia system, and open hypermedia. This work discusses experiments that were conducted to assess the benefit of integrating Semantic Web technologies with open hypermedia and adaptive hypermedia concepts to provide adaptable hypermedia for users' requirements. In this work, ontologies are used to explicitly define models, concepts, user profiles, context and semantic relationships. To enable sharing and reusability of information chunks as annotations, this research brings together several technologies: ontologies for knowledge representation, and extended FOHM to represent the structure of annotations. A contextual annotation framework (CAF) using an ontology-based contextual annotation service is proposed.

The novel contribution of this work is introducing the CAF using an ontology-based contextual annotation service by building on open hypertext and the Semantic Web. The a-PIE is prototyped to provide a system for supporting browsing, reading, annotating hypertext, and manipulating interesting annotations in a personal repository. The framework has been applied to the specific domain of web development, in order to carry out a focused evaluation. The results indicate that the framework is valid.

Contents

List of Figures

List of Tables

Acknowledgements

First, I would like to thank my supervisors Professor Wendy Hall for her constructive supervision and for all her support, and Dr Gary Wills for his continuous excellent supervision, support, guidance and understanding of my abilities.

As importantly I thank my family – beloved parents, two sisters, and cousins for all their never-ending support and encouragement.

Special thanks go to Quintin for his expert English in proof-reading of my thesis. I have learned new phrases, expressions and grammar structures.

I would like to express my gratitude to members of the IAM, LSL, colleagues at the National Statistical Office and WS. Many thanks to my friends many of whom I have made along the way, Dr. Christopher Bailey, Dr. Jing Zhou, Dr. Nor Aniza Abdullah, Dr Panchit Longpradit, Dr Waraporn Jirapanthong and Dr Srisupa Palagawongsa, Thai friends and all of my friends who have supported me, have helped me, one way or another, in the completion of this thesis.

Others, who I did not mention here, but I always appreciated their kindness.

Finally, I would like to thank the Royal Thai Government for continuing to fund my study in UK. I have had the opportunity to work with and learn from wonderful people.

Thank you all.

Chapter 1 Introduction

1.1 Research Overview

"*Knowledge is itself power*" Francis Bacon (1561-1626) – Of Heresies

Knowledge is an essential asset for any organisation (Benjamins, *et al.*, 1998). Knowledge can be tacit (in the human mind) or explicit, documented or procedural, for example, or it can be information, experiences or skills which can be managed through technology (Hildreth and Kimble, 2002). According to Nonaka, explicit knowledge (or information) is knowledge that is easily expressed, captured, stored and reused. Explicit knowledge can be transferred from tacit knowledge, and transmitted as data and is found in databases, books, manuals and messages (Nonaka, 1991). Al-Hawamdeh mentioned that knowledge in books or journals will be useful and usable when it is read, manipulated and communicated from one person to another (Al-Hawamdeh, 2002). Knowledge management refers to managing the knowledge, like organising memories, expert systems, document management systems or learning organisations (Wiig, 1994). However, "knowledge acquired through experience cannot get reused because it is not shared in a formal way" (Angus, *et al.*, 1998). Knowledge management is the process of knowledge gathering by adding value to previous knowledge (Duffy, 1999). These processes have to be implemented over a period of time and need human cooperation through technology (Benjamins, *et al.*, 1998) (Al-Hawamdeh, 2002).

In a community of practice people are often willing to develop and share knowledge. They could use a repository for this purpose, but sometimes the context and relationship between information chunks will be lost. Knowledge management and its associated tools aim to provide such an environment in which people may create, learn, share, use and reuse knowledge, for the benefit of the organisation, the people who work in it, and the organisation's customers. However, instead of helping users, many systems just lead to increasing the cognitive overhead (Conklin, 1987a). Adapting the links to document content to the needs of individual users greatly enhances the navigation and comprehension of an information space, and reduces the information overload.

Learning is often social and is more effective when it takes place in groups, especially when groups collaborate directly and they teach each other. Groups are important structures of any organisation or community where reflection is concerned. Virtual communities as communities of practice (CoP) are established for groups of people who would like to share knowledge and have common interests in similar topics. These CoP may be groups of people that come together as a formal or informal network. People in a CoP network can enrich their knowledge and expertise in a certain domain through interaction. This enables them to build on knowledge and contribute it to each other (Seely Brown and Duguid, 1991).

Another definition of CoP is an informal group of people who are interested in the same topic of knowledge or competence who are willing to share and learn together to develop that knowledge through interaction in an environment where knowledge is created, nurtured and sustained (Library, 2005), (Hildreth and Kimble, 2002). Community of practice preserves the tacit aspects of knowledge that formal systems cannot capture (Wenger, 1998). It is noted that knowledge can be shared through narration and story-telling by members (Easterby-Smith and Lyles, 2003) and there are several ways that technologies can support the community of practice (Wenger, *et al.*, 2005).

Knowledge management tools should provide users with the ability to organise information into convenient information chunks. Relationships within the stored information are necessary. The knowledge management tools should support the better use of the available raw information by providing facilities to search and abstract the information, assist users to share new information, and also present it to users in new ways (Davies, *et al.*, 2003a). Instead of helping users, many systems are increasing the amount of unrelated information returned which makes it difficult for users to find, organise, access and maintain the information.

Hypermedia is an extension of the term hypertext, in that several media such as text, graphic, video are connected to create related information. It enables users to traverse in a non-linear manner from one node to another node through links, called hyperlinks (Bieber, 2000; Lowe and Hall, 1999).

Adaptive hypermedia research, a sub-discipline of hypermedia, has aimed to improve methods and techniques that reduce cognitive overhead and disorientation problems by providing information to individual users, based on contexts. Contexts are conditions or characteristics, such as the users' profiles, their interests, the environments or devices, used in a system.

Open hypermedia research is also an area of hypermedia. The principle of this research is separating the links, instead of embedding them as in typical hypermedia, out of the source document and manipulating these links as first class objects (Wills, *et al.*, 2003).

A key idea of a Semantic Web is to have data defined and linked in such a way that its meaning is explicitly interpretable by software processes rather than just being implicitly interpretable by humans (Berners-Lee, *et al.*, 2001). An ontology is a means of describing the shared understanding of a domain, and its use allows the capture of knowledge and facilitates knowledge sharing and reuse (Gruber, 1993). Ontology can be defined using the semantic web languages.

As noted by Marshall, annotation is a method in which hypertexts grow. It is a fundamental activity for hypertext readers. Annotation can reduce the cognitive overhead to users when they retrieve an annotated document because users can spot the annotated area (Marshall, 1998a). Annotation tools can be used as a knowledge management tool by assisting users to build personalised document summary to see what was important in the document and, share and learn from annotations when annotation tools are used in a community of practice. Annotation systems should provide relevant annotation to users by enabling users to specify the context in which they want to see the annotations. This will address in the thesis.

1.2 Research Motivation

> *"The reason I encouraged such annotations was that I remembered that when I was in college with Ted [Nelson], I would always grab the dirtiest copy of a book from the library, rather than the cleanest one, because the dirtiest ones had the most marginalia, which I found helpful."* (van Dam, 1988).

Marshall pointed out a number of research issues related to annotation: While some people think that some of the annotations are valuable, some think that some annotations are annoying. Some users would like to add keywords to each annotation, such as people, places, things, or geographical regions, in order to make their annotations accessible easily over the long term. Annotations might be tied to situational factors as contexts. For example, a lawyer annotates the same case differently for different uses (Marshall, *et al.*, 2001). Annotations can be in private and public forms for different situations. Annotations users make in class that captures what the professor is saying are considered more important than what students have made in their own readings of the material (Marshall, 1998a). Since Wolfe pointed out in her study of how students value the annotations of experts, the source of the interpretation is very important (Wolfe, 2000).

Annotation systems should consider the interoperability issue, because sometime users would like to annotate different types of media and manage these annotations in a single repository. Users would like to organise, re-order or categorise annotations, for example, lawyers classify some annotations at ‘pro’ cases (support arguments) or ‘con’ cases (present counterarguments) (Marshall, *et al.*, 2001).

A scenario will help explain the motivation behind this research. Suppose users in a virtual community are reading web documents on an e-learning web site. They find an interesting piece of information and would like to keep it for reference later, in a personal repository. Occasionally, they would like to add annotations to particular information snippets and would like to record what type of annotations of the information snippets are; such as comments, examples or explanations. The users annotate the information snippets in a way that is relevant to them for storage in their personal repository. Sometime they would like to record one or more criteria for each annotation, for instance keywords or difficulty level before storing it in their repository. As in any collaborative environment, they would also like to share these annotations with others like social learning environment. The users can also learn and gain knowledge from others’ annotations. As this is a diverse community, users have different levels of expertise so when reading someone else’s annotation they would like to know what the level of difficulty associated with each annotation was. At a certain time they are interested in only some topics, so they want to see annotations of these specific topics instead of all annotations. In addition, users want to store others’ annotations into their repository and sometime they want to set their own contexts at the annotations for archiving later. Subsequently, depending on the reader’s profile, only those authorised by the original user will see the information and its annotations.

Therefore, to satisfy the scenario stated above the users must be able to:

- browse web documents and augmented these with the relevant annotations.
- add context to an annotation.
- keep annotations in a personal repository.
- share annotations with others.
- view annotations based on their personal profiles.

For supporting communities of practice, a web-based system is important since it provides annotations personalised to the needs of individual users, and the functionalities of knowledge sharing and reuse. A personalised web-based system greatly enhances navigation and comprehension of information spaces, while knowledge sharing and reuse facilities offer functionalities for users to manipulate their personal information repository, and to annotate particular information prior to passing it to others.

1.3 Research Objectives

According to the research motivation presented in Section 1.2, the objective of this work is to investigate how semantic web technologies, open hypermedia and adaptive hypermedia principles can be used to provide appropriate annotations for users in web content. In addition, how annotation with context systems can be able to support the shareability and reusability of annotations has been examined. The new framework has been proposed and presented in this thesis.

The research question of the thesis is whether it is beneficial to provide contextual annotations in web-based applications.

The research result was assessed by both practical evaluation and by theoretical comparison between existing systems.

1.4 Document Structure

This thesis describes the process of developing an adaptable annotation framework. While early chapters document existing research and the initial work of the author, later chapters are concerned with the rationale, design, evaluation and discussion of the CAF.

Chapter 2 presents the field of hypertext and adaptive hypermedia. The adaptive hypermedia system components are documented. The various techniques of adaptive hypermedia are presented together with a selection of sample frameworks, including an examination of existing frameworks for adaptive hypermedia.

Chapter 3 describes the field of open hypermedia, a topic that originated from hypertext research at the end of 1980's. Open hypermedia research concentrates on the issues concerning existing hypermedia systems. The developments of link services are produced as a new ways of expressing hypermedia systems.

Chapter 4 documents the field of semantic web technology. Some examples of web-based annotation systems are introduced and their characteristics are examined.

Chapter 5 presents the initial works by the author in these fields. Two applications have been prototyped using the Auld Linky link service with the semantic web technologies for e-learning application and adaptable personal information environment.

Chapter 6 details the development of the CAF. The early sections state the need for a new framework and identify the limitations of existing approaches. This is followed by a discussion of the design of the CAF approach showing how CAF unifies open hypermedia, adaptive hypermedia principles and semantic web in the framework, goes on to describe the implementation of the framework as a web-based annotation application to drive a tour of the framework features. A more detailed description of the software implementation is also provided before discussing how the CAF responds to the research issues. This chapter concludes with an analysis of some of the advantages that this framework provides, such as a new service, a flexible and extensible approach to system development, and an open architecture for future annotation systems.

Chapter 7 evaluates the CAF in practice and follows with a theoretical discussion on the characteristics of existing approaches and the CAF.

Chapter 8 presents the conclusions of this research and future work.

1.5 Declaration

This thesis describes the research undertaken by the author while working within a collaborative research environment. This documents the original work of the author except in the following sections.

The Auld Linky a contextual link server originally developed as part of the EQUATOR project by Dr David Millard, Dr Danius Michaelides and Dr Mark Weal (Michaelides *et al.*, 2001). The proxy server, developed by Bailey (Bailey, 2002), was modified by author to be SWLP proxy server which was used in the initial work in Chapter 5.

1.6 Publications

Maneewatthana, T., Wills, G. and Hall, W. (2007) *Adaptive Personal Information Environment based on the Semantic Web.* Journal of Digital Information, 8 (3). ISSN 1368-7506.

Maneewatthana, T., Wills, G. and Hall, W. (2006) Adaptation in Adaptable Personal Information Environment (Poster). In *Proceedings of Adaptive Hypermedia and Adaptive Web-Based Systems*, Dublin, Ireland.

Maneewatthana, T., Wills, G., and Hall, W. (2005) Adaptive Personal Information Environment based on the Semantic Web. In *Proceedings of Sixteenth ACM Conference on Hypertext and Hypermedia*, Salzburg, Austria., pp 214-216.

Maneewatthana, T., Wills, G., and Hall, W. (2005) Adaptive Personal Information Environment based on the Semantic Web. In Proceedings of Hypertext 2005 - ACM International Workshop on Adaptive and Personalized Semantic Web, Salzburg, Austria, pp 65-73.

Maneewatthana, T., Wills, G. and Hall, W. (2004) Adaptive Link Services for the Semantic Web (Poster). In *Proceedings of 14th International Conference on Knowledge Engineering and Knowledge Management (EKAW)* 3257 / 2004, pp. 469-470, Whittlebury Hall, UK. Motta, E., Shadbolt, N., Stutt, A. and Gibbins, N., Eds.

Chapter 2 Hypermedia

2.1 Introduction

The previous chapter introduced motivation for this research. This chapter presents the concepts of hypertext/hypermedia, adaptable, personalised and adaptive hypermedia, and also describes adaptive hypermedia techniques. Since this work focuses on describing a framework for personalised hypermedia, the models and frameworks for hypermedia are also presented.

2.2 History of the Web

> "*The human mind … operates by association…. Man cannot hope fully to duplicate this mental process artificially, but he certainly ought to be able to learn from it… One cannot hope to equal the speed and flexibility with which the mind follows an associative trail, but it should be possible to beat the mind decisively in regard to the permanence and clarity of the items resurrected from storage.*" (Bush, 1945)

There is a lot of information available and it continues to grow at an exponential rate. Information plays an important role in daily lives and work. For this reason, people need to create a personalised subset of the information that they can use when they need (Bruce, 2005). In 1945, Bush had a vision for a personal collection of information that an individual could access anytime. He described hypertext in the article "*As We May Think*" introducing Memex as a machine that could make notes and links between two points of objects i.e. books, records, in order to support indexing and retrieval (Bush, 1945). This idea influenced Engelbart's idea of augmentation, as appeared in "*Augmenting Human Intellect: A Conceptual Framework*" (Engelbart, 1963). Later, in 1968 the first successfully implementation of hypertext was implemented, called oN Line System (NLS) (Engelbart and English, 1968), by the Augmented Human Intellect Research Center at SRI. NLS was a groupware system assisting creation of digital libraries, and storage and retrieval of electronic documents using hypertext. Files in NLS supported both *reference links* and non-hierarchical links. Files in NLS were structured in a hierarchy of segments which were called "statements". Each file had an identifier of its level and could be linked or referred to within or other files (Conklin, 1987a; Conklin, 1987b).

While Engelbart was developing the augment system, Ted Nelson developed his own ideas about the creation of a universal electronic library where any text of any document could be linked to any text of another other document (Nelson, 1965). In 1965, Ted Nelson coined the term "*hypertext*" for "*non-sequential writing*" (Conklin, 1987a). It means that hypertext refers to a document in which objects i.e. annotations, footnotes, or documents are linked associatively in a complicated way.

In the following years, several hypertext systems were developed and hypertext systems evolved into *hypermedia* systems. The term "*hypermedia*", created by Ted Nelson, was defined as an extension of hypertext, in which any medium such as graphics, a piece of text, or video were connected non-linearly as a collection of interrelated nodes (W3C, 1992). Nowadays, the words hypermedia and hypertext are used interchangeably. In 1990, Tim Berners-Lee demonstrated the concept of the hypertext client-server approach in a remote and distributed environment at CERN which led to what today is known as the World Wide Web (Berners-Lee and Cailliau, 1990). The World Wide Web is like an encyclopaedia, a collection of records or even telephone directory, for example, which can be accessible through any computer (Gillies and Cailliau, 2000). A comprehensive survey of hypertext and early hypertext systems and applications can be found in (Conklin, 1987a; Conklin, 1987b).

2.3 Hypertext/Hypermedia

Human memory is associative, while hypertext is a means to mimic writing and processes, and represents what is inside a human's brain (Lowe and Hall, 1999). As mentioned in the previous section, hypermedia is a network of information elements that are interconnected associatively or in non-linear structures, and users can navigate freely using hyperlinks. Information elements can be any piece of information or media such as video, image or a piece of text. Hyperlink is a reference or navigation element in a document to another part of the same document, another specified part of another document or to another whole document (Bieber, 2000; Lowe and Hall, 1999).

According to Bieber hypertext components are composed of *nodes*, *links*, *link anchors*, *link markers*, and *composites* (Bieber, 2000). Nodes contain content and attributes of information objects. Each node should store a single chunk of information or only one idea. Composite nodes are links of a set of nodes, for instance, a book with many chapters. Nodes can have semantic types displayed with the node in order to support structural search for a particular node. Links connect entire nodes, while link anchors specify a particular part within each node. Link anchors can contain internal parameters, for example how to find its link. However, normally users do not see the link anchors as they are embedded in the code. A link marker displays the area that user can select to activate the anchor and traverse its link. Yee describes an anchor which addresses an entire document as a coarse-grained link, whereas an anchor which addresses a specific part of the document he calls a fine-grained link (Yee, 2002). URLs (Berners-Lee, *et al.*, 2005) support both kind of anchors (using <LINK> and <A> tags). The HTML specification allows one-way links with fine-grained origin anchors and coarse-grained target anchors. HTML links are intrinsic, meaning that they must be embedded in the linked document itself (Yee, 2002).

Links can represent relationships between nodes, as *link types*, and can have *link behaviours* attached (Bieber, 2000). The simple node and link model is not rich and not enough to provide functionalities for managing and representing information required by many applications (Halasz, 1988). HTML (Berners-Lee and Connolly, 1995) provides the REL and REV attributes on hyperlinks for specifying the relationship between the two anchors. Therefore, Bieber (2000) noted that links may have *semantic link types* and associated *keywords* which can guide or help users decide whether to traverse links by showing how nodes are related. *Link behaviour* identifies which action needs to be executed when users activate that link, for example, activating an “execute” link would run a program. Links could be treated as first class objects and stored in external link databases (or *linkbases*), and manipulated using a *link service* (the concept of linkbases and link service will be described in detail in Chapter 3).

Several taxonomies of links have been proposed in order to analyse how they are best utilised. Described by Lowe and Hall (1999), one taxonomy could be based on the mechanics of the links, by considering at the number of sources and destinations for links (for example, single-source single-destination, multiple-source single-destination), or the directionality of links (uni-directional, bi-directional), and the anchoring mechanism (such as generic links, dynamic links). The web currently supports links joining two anchors: an origin (the location with <A> tag) and a target (the location identified in the HREF attribute). Therefore, hyperlinks on the web can be followed only from the origin to the target. On the other hand, bi-directional links are links that can be followed in both directions. In addition, a link taxonomy can be created based on the type of information relationships being represented. Moreover, links can be classified based on the organisation of the information space (structural link) and those links related to the content of the information space (associative and referential links). A structural link is a linear or hierarchical structure without the semantic relationship between linked information. An associative link is independent from the structure of the information, and it represents the semantic relationship between information elements. An associative link can attach to any piece of a document and it can be organised or created on the fly. De Rose claimed that an associative link cannot be replaced by a retrieval algorithm (De Rose, 1989). A referential link provides a link between an item of information and an explanation or definition of that item; more specifically both of them have to be exist (Lowe and Hall, 1999).

Following (Grønbæk and Trigg, 1999), Whitehead describes four classes of link styles in hypertext systems (Whitehead, 2000). These are: links as addresses (address of the link destination is embedded within the code, i.e. WWW), links as associations (links are first-class objects that express an association between entities, as implemented in Chimera (Anderson, *et al.*, 2000), Intermedia (Yankelovich, *et al.*, 1988), FOHM (Millard, *et al.*, 2000)), links as structural elements (links are used to represent hierarchical or organisation of materials), and links for rhetorical representation (links represent the structure of an argument). Whitehead added a fifth style which was links as semantic network (link types are used to represent semantic relationships between works, and may be designed for link navigation such as the Semantic Web (Berners-Lee, *et al.*, 2001). Whitehead also surveyed several systems to develop terms used to describe the notions of work, anchor, and link in hypertext systems (Whitehead, 2000).

Yoo and Bieber proposed the link taxonomies for generic relationships considered within a generic information system. The relationships among items of interest which can be objects, people, commands or even pieces of meta-information are defined (Yoo and Bieber, 2000).

When making links authors can enhance the information space with context which produces alternative designs and enables users to freely access the enriched information for better understanding. In addition, hypermedia enhances human comprehension because the hypermedia mimics an associative network and this behaves in a similar way to human cognition in retrieving and storing information (Conklin, 1987a; Yoo and Bieber, 2000).

There are several operational advantages of hypertext. For example, it is easy to structure information, meaning that we can organise or aggregate elements with relationships to each other. It is easy to locate or create new references. Users are able to develop their own networks, or add some information such as a note or a comment to another's document while the original document is left unchanged. Users can view the global structure and then restructure or customise documents. The same parts of documents can be pointed to by many references which supports the concept of modularity to reduce overlap and duplication (Conklin, 1987a).

There is huge amount of information available on the web and it keeps growing. As a result of that, hypertext causes two problems for the user: *lost in space* (Conklin, 1987a) or *disorientation* (Bieber, 2000), and *cognitive overhead* (Conklin, 1987a) or *cognitive overload* (Bieber, 2000) or *information overload* (Nielsen, 1995), or *knowledge overload* (Bray, 2007). Lost in space occurs when users feel like they do not know the current state or location in the system and have difficulty finding where the information they need is, or deciding where to navigate next (earlier or later in the text) (Bieber, 2000; Conklin, 1987a; Nielsen, 1995). Users can lose their orientation even in small document (Nielsen, 1995). However, several possible solutions to the navigation problem have been researched such as using trails (Nielsen, 1995), graphical browsers, or search and query techniques (Conklin, 1987a). Cognitive overhead is defined as "the additional effort and concentration necessary to maintain several tasks or trails at one time" (Conklin, 1987a). It happens when users are given too many data links to follow and they need to make a decision as to which one to follow (Bieber, 2000; Conklin, 1987b; Carlson, 2003). Information overload is also indicated as a problem that people suffer where there is more irrelevant data than they can absorb (Ackoff, 1967). Cognitive overhead has been associated with the limited capacity of the human short-term memory (Head, *et al.*, 2000). In a web environment, we should consider how to facilitate recognition rather than to force limited recall. Cognitive overhead can be approached by research on information filtering techniques and improvements in performance and design of hypertext systems (Conklin, 1987a; Head, *et al.*, 2000). One example is to provide a type and a label on the links as implemented in NoteCards system (Halasz, *et al.*, 1987). In addition, Bieber noted that these two problems could be resolved by using: good user interface design principles, semantic nodes and link types, filtering based on user task and preferences, and hypertext navigation, annotation (including bookmarks, landmarks and comments), and structural features (Bieber, 2000).

2.4 Adaptive, Adaptable, Personalised Hypermedia

The web is growing very fast, providing vast quantities of information. As a result of this success, its biggest problem is that of information overload (Gillies and Cailliau, 2000). In addition, there is so much information available on the web, it is difficult to find what we want. As mentioned in last section, the shortcomings in using the web to publish information are cognitive overhead and user being lost in hyperspace. While every user needs different information, they have different preferences, goals, backgrounds, knowledge and abilities. Therefore only hypermedia systems that only offer the same set of links or hypermedia to all users are not enough. It has been suggested that Adaptive Hypermedia (AH) systems can be beneficial to overcome these problems where the hyperspace is large and the system might be operated by users with different goals and knowledge, who therefore need different information on the same topic (Bailey, 2002; Brusilovsky, 1994; Brusilovsky, 1996b; Conlan, 2004). In this case, a user model is one of the most important components as it will capture information about users in order to provide relevant information about them.

In AH systems the distinction between adaptive, adaptable and personalise are defined in the Oxford English Dictionary as:

> Adapt means "*to change something in order to make it suitable for a new use or situation*".
>
> Adaptable means "*able to change or be changed in order to deal successfully with new situations*".
>
> Adaptive means "*(technical) concerned with changing; able to change when necessary in order to deal with different situations*".

> Personalise means "*to design or change something so that it is suitable for the needs of a particular person*". (Hornby, 2000)

Brusilovsky gave a functionality-oriented definition of AH as:

> "*By adaptive hypermedia systems we mean all hypertext and hypermedia systems which reflect some features of the user in the user model and apply this model to adapt various visible aspects of the system to the user. In other words, the system should satisfy three criteria: it should be a hypertext or hypermedia system; it should have a user model; it should be able to adapt the hypermedia using this model.*" (Brusilovsky, 1996a).

Hothi noted that *adapted hypermedia* systems are those which can be adapted to various groups of end users during design, while *adaptable systems* allow end users to change the functionality or characteristics of the hypermedia systems. Hypermedia systems can be adaptive if they adapt dynamically during the session according to end users' behaviours (Hothi, 2000).

Here are some other definitions. If a hypermedia system attempts to gather users' information in user model and provide information suitable to particular users' needs or properties based on that user model, then it is called an AH system, while a system that allows users to select from a range of parameters according to their needs is called an adaptable system (Wadge and schraefel, 2001) (schraefel, *et al.*, 2003). The adaptable system would be useful in the cases where users cannot make a decision, or cannot specify a well-defined goal, or their goals may change throughout exploration, or moreover, or their expertise changes from one section to another (schraefel, *et al.*, 2003). If the system provides options for users to select or decide through any configuration form or questionnaire, then it is an adaptable system (De Bra, 2000). Brusilovsky mentioned that since 1996, goals/tasks, knowledge, background and preferences were modelled and used for making adaptation decisions by several systems (Brusilovsky, 2001). AH systems can deliver personalised views of hypermedia documents to users (Bailey, *et al.*, 2002; Brusilovsky, 2001). By perceiving the users' behaviour, it is possible to perform the adaptation automatically (De Bra, *et al.*, 1999).

According to Sheth, personalisation can be classified in three ways: cognitive, economic and social. Cognitive systems choose documents based on characteristics of their contents. Economic systems select documents according to some value calculation, while social systems select documents based on the recommendation of other users (Sheth, 1994). The application of social and cognitive systems relies on the application area. A social system is suitable for up-to-date information within a community, while a cognitive system is appropriate for a topic-specific information requirement.

Based on the definitions presented above, we can conclude that personalisation is the result of an adapting contextual elements to users (Abdullah, 2006; Conlan, 2004). In other words, the objective of personalised information systems is giving the individual user optimal support in accessing, retrieving and storing information. Examples of user requirements are the current task, the goal of the user, the context in which the user is requesting the information, the previous information requests and the level of expertise (Baldoni, *et al.*, 2005). The distinction between personalisation actions and adaptive actions is that the former are any action that alters the structure of a hyperpage activated by a user, while latter are system-initiated personalisation actions (Ohene-Djan, *et al.*, 2003). However, it was suggested that users should be able to exercise control over the adaptivity but should not have to control it continuously (Espinoza, 1996).

Adaptive hypermedia can be used in several application areas: educational hypermedia, online information systems, online help systems, information retrieval hypermedia systems, and institutional information systems. There are three application areas that are currently popular. First, online documentation systems use adaptive hypermedia in order to provide different information for different users and provide individualised navigation support in a huge information space. Secondly, application-specific and system-specific applications with advanced help and explanation facilities adopt adaptive hypermedia in order to adapt explanations to different users in system details and concepts, such as a decision support system. Thirdly, educational systems use adaptive presentation and navigation to support student-driven exploration of educational material (Brusilovsky, 1996b; Brusilovsky, 2001).

There are several aspects of a user that hypermedia systems are able to adapt to. Some examples are now given. One of the classical approaches is to allow adaptation to be based on cultural background such as natural language, familiar weights and measures or specific styles of writing. Another classical approach from the Human Computer

Interaction (HCI) field is to provide a user interface adapted to the user's preferences. Another aspect is the communication style and needs of users, since users differ in their communication styles such as those needing clear direction and those needing a broader freedom of choices. In addition, some users have special communication needs, for instance handicapped users. Prior knowledge of users can be used to determine which information to provide. Education systems typically use this aspect. Learning history can also be considered. Users with different background or knowledge may have different ways of acquiring knowledge within domains, thus explanation of content and navigation support should be adapted appropriately. For example, the system might provide clear direction for beginners, while more freedom for experts. Another aspect is the aims and goals of users. In education, for example, users with different roles, teacher and learner, may have different conceptions about the aims and goals of learning. Therefore a system should adapt information by directing learners towards content that teachers have specified as a goal (Conlan, 2004).

It was explained by De Bra that adaptive hypermedia applications are different from non-adaptive applications (De Bra, 2000). There are both advantages and disadvantages of offering adaptive hypermedia techniques. First, in adaptive systems information can be presented at the right level of difficulty, in the right medium and with the right writing style for each user. In order to provide these, all of different versions of contents or pages must be authored, including the sequence and the relationship of each fragment to its concepts also needs to be determined. The AHAM model (described in Section 2.7.2) provided this since it contained a domain model. Secondly, users can be guided towards information that is relevant for them at that moment. In this case, if guidance takes users on the wrong path or omits prerequisite relationships to irrelevant pages, users might not understand. Therefore, bad guidance is worse than no guidance. Thirdly, users might get confused because of unstable appearance of pages, meaning that each time a user revisits a certain page, the contents or links look different because the document has been adapted according to the user model.

Hypermedia consists of a set of pages connected by links. A page contains information and links which can be connected to locations in the same page or different pages. In order to provide adaptive hypermedia, there are common components typically found in an AH system. In AH, what can be adapted are the content of a page and the links to related pages (Brusilovsky, 1996b). The adaptive hypermedia techniques and system components will be described in the next sections.

2.5 Techniques of Adaptive Hypermedia

There are several methods for providing the adaptivity in hypermedia systems. Brusilovsky's taxonomy (Brusilovsky, 2001), which classified adaptation techniques that are applied in AH systems is shown in Figure 2-1. Brusilovsky described the two main types of adaptation technique as adaptive presentation and adaptive navigation.

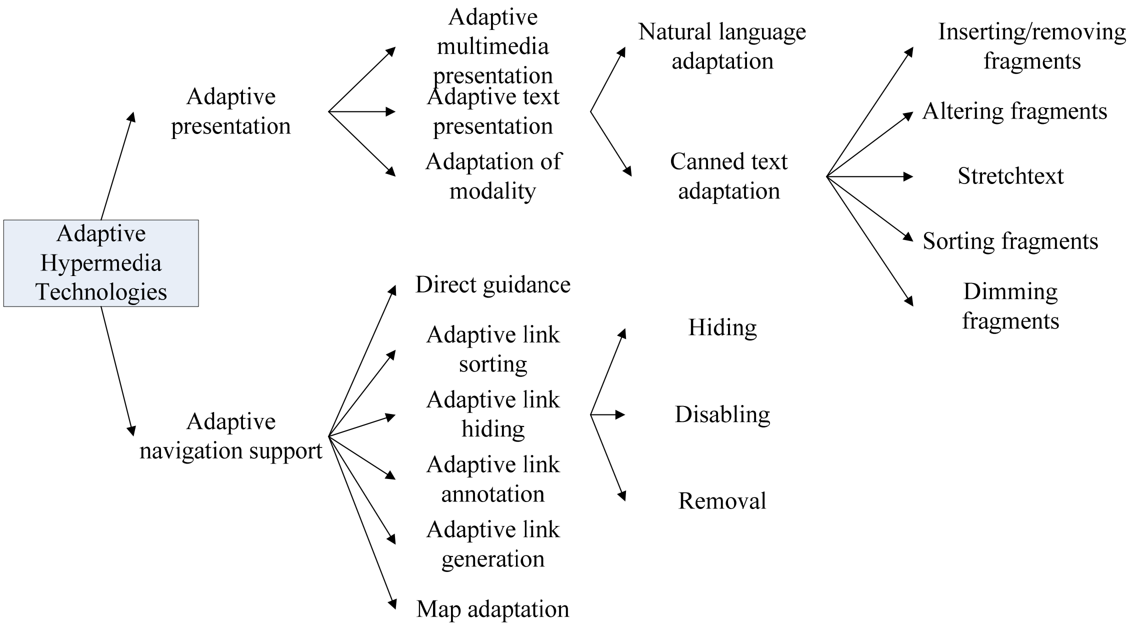


Figure 2-1 The updated taxonomy of adaptation hypermedia technologies (Brusilovsky, 2001)

2.5.1 *Adaptive Presentation*

Adaptive presentation is altering the content of a page presented to a user according to that user's knowledge, goals or other characteristics. The content of a page can consist of not only text but also multimedia objects such as images or audio. Adaptive presentation can be adaptive multimedia presentation, adaptive text presentation, and adaptation of modality. Adaptive presentation is primarily used to lessen information overload within hypermedia pages. The most popular method used is altering (hiding/showing) part of the information depending on its relevancy to the user's knowledge or interests. The main disadvantage of the altering fragment technique is that the user might be deprived of information which might be beneficial. However, the stretchtext method can resolve this problem. The major problem with adaptive presentation systems is that a very high level of domain understanding is needed in order to change the presentation of concepts in the contents. However, this problem can be reduced by using content abstraction to attain domain independence (Bailey, 2002; Brusilovsky, 1996a).

2.5.2 Adaptive Navigation

Adaptive navigation is manipulating the order and the appearance of the links in a way that would assist the users in seeking the most valuable and pertinent information (Bailey, 2002; Brusilovsky, 2001). Brusilovsky's taxonomy specified six main types of navigation support: direct guidance, adaptive link sorting, adaptive link hiding, adaptive link annotation, adaptive link generation and map adaptation. These groups can use individually or can be combined to provide navigation support. These methods are briefly described below.

- *Direct guidance*: the system decides the next links for a user, based on the user model.

- *Adaptive link sorting*: hyperlinks are sorted with the topmost items having the most relevance to a user. The weight of the links is based on the user model.

- *Adaptive link hiding*: hyperlink appearance is made invisible if it is not relevant to the user. This can reduce information overload for the user.

- *Adaptive link annotation*: a hyperlink's text style or appearance is changed according to the user model. The hyperlink linking to a relevant page will be annotated with visual cues; for example, special colour, icon, or font size.

- *Adaptive link generation*: additional links which are leading to useful or relevant information for a user are added to an existing page.

- *Map adaptation*: alter the form of hypermedia for a user by using other adaptive navigational techniques.

In addition to adaptive navigation support in Brusilovsky's taxonomy, Bailey introduced link augmentation (Bailey, *et al.*, 2001) as another adaptive navigation technique, from their research in the fields of open hypermedia and adaptive hypermedia fields. Link augmentation is defined as "*a technique whereby external links are inserted directly into the body of a document*". The difference between link annotation and link augmentation is that link annotation process relates to the visible properties of links while link augmentation relates to adding more links to existing hyperdocuments (Bailey, *et al.*, 2001).

2.6 Adaptive Hypermedia System Components

An adaptive hypermedia system, described by Wu, normally comprises a domain model, a user model, an adaptation model and an adaptive engine (Wu, *et al.*, 1998).

- *Domain model* is a set of domain concepts along with their relationships, describing how information content of the application is structured. It can be used to describe knowledge contained in hypermedia document. De Bra described three kinds of concept: atomic concepts or fragments, pages (composed of atomic concepts) and abstract concepts (representing larger units of information) (De Bra, *et al.*, 1999). Each concept is connected through concept relationships such as prerequisite relationships. There are many methods that have been used to structure a domain model, for instance linear, concept graph, semantic network, hierarchical tree and combined structure (Carro, 2002).

- *User model* stores the individual's characteristics, for instance preferences, knowledge, goals, navigation history, and other relevant aspects. A user model plays an important role in determining the success of the adaptation process. User models may be used to classify all pages into many groups based on the user's current knowledge and interests or goals (De Bra, *et al.*, 1999). In AH systems, user models are usually based on domain models (Brusilovsky, 1996b). Kavcic described three important aspects that have to be considered when designing a user model. First, the types of user's information that needs to be captured and how it may be obtained. Secondly, how the information in the system can be represented, and thirdly, how the model can be constructed and updated (Kavcic, 2000). Another way of looking at information captured in a user model is to divide it into two types: static and dynamic information. By static we mean that the system will exploit user information when it is first created during the interaction. On the other hand, by dynamic we mean that it will gather user information while the interaction is in progress and use this information to update the user model continuously (Hothi, 2000; Kavcic, 2000). The user profile can be captured using two main methods; the overlay model and the stereotypical model. Brusilovsky noted that one of the powerful and flexible methods used to model user knowledge is the overlay model, since it can assess independently user knowledge on different topics. The model of user knowledge can be represented as a set "topic - value" pairs. This can also be used to model a user's goals, background or other personal details. On the other hand, the stereotypical model is simpler as it assumes that all users will be classified into a specific class, for example, beginner intermediate and expert (Brusilovsky, 1996b). According to Allen, user models have three different dimensions: long term/short term, individual/group, and explicit/implicit dimensions. The long term/short term dimension handles collection and persistence of information over a period of time. The individual/group dimension considers whether the information is applied to all users in a common group or only to an individual user. Lastly, the explicit/implicit dimension specifies how the model is pulled out directly or indirectly from user behaviour (Allen, 1990).

- *Adaptation model* normally contains rules that define how the domain model relates to the user model to specify adaptation. The most common form of adaptation model is "if <condition> then <action>" (Wu, *et al.*, 1998).

- *Adaptive engine* will do the action (i.e. presentation of links or fragments of hypermedia content pages) when the condition in adaptation model is true.

There are several formal models which have been proposed that focus on the mechanisms and structures based on adaptive hypermedia systems. The next section will present some models and frameworks in hypermedia.

2.7 Models and Framework

The AH research community has produced applications operating in various domains, for example in electronic shops (Ardissono and Goy, 1999), in collaborative recommendation (Balabanovi and Shoham, 1997), and e-learning systems. According to Bailey models describe the components and descriptive languages used for specifying generalised hypermedia systems, while frameworks represent real systems that the authors claimed that they can be used in a variety of applications (Bailey, 2002). In practice, adaptive hypermedia systems are usually limited to a specific application area, for instance an adaptive learning environment. Here are some models and frameworks of hypermedia and adaptive hypermedia.

2.7.1 Hypertext Abstract Machine

The Hypertext Abstract Machine (HAM) is a general-purpose, transaction-based, server for a hypertext storage system (Campbell and Goodman, 1988). It is a low-level storage engine and can be used in several different hypertext applications. The hypertext system architecture based on HAM contains the following layers (Figure 2-2).

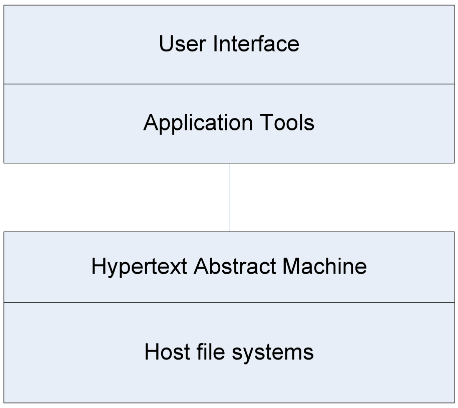


Figure 2-2 Generic Hypertext Abstract Machine (Campbell and Goodman, 1988)

- *User interface*: A presentation level contains window-based interactive environment for applications to communicate with users.

- *Application tools*: The actual applications which may or mat not run on the same machine as the HAM.

- *Hypertext Abstract Machine*: An engine which manages all information about the hypertext and communicates with the application through a byte stream protocol.

- *Host file system or storage system*: A repository which stores all the hypertext graphs or databases.

The HAM storage model consists of five major objects: graphs, contexts, nodes, links, and attributes. Graphs are networks of nodes and links containing one or more contexts. Contexts are partitions of data within a graph which could be used for several purposes, for instance, to support configurations. Nodes are data which could be text or fixed-length binary blocks. Links are relationships between a source and a destination node with bi-directional linking. There is also a cross-context link which relates two nodes in different contexts and is useful for sharing data between two contexts. Attributes can be attached to contexts, nodes and links. Attributes-value pairs contain semantics describing HAM objects.

The following operations can be performed on HAM objects: create, delete, destroy, change, get, filter, and special. The filtering mechanism allows a user to determine visibility by specifying subsets of HAM objects to be extracted from large graphs. The HAM architecture provides version control, filtering and data security. The HAM storage model has been successfully tested against systems such as Guide (Brown, 1987), Intermedia (Garrett, *et al.*, 1986; Yankelovich, *et al.*, 1988), and NoteCards (Halasz, 1987; Halasz, *et al.*, 1987).

Guide was developed by Peter Brown as a research project at the University of Canterbury. It became a commercial product later. It is a tool for writing and reading electronic documents. In this system, text and graphics are integrated together in articles or documents. Guide supports four different kinds of link: replacement buttons (a button itself contains a further button for a greater level of detail and allows the user to go back and forth), note buttons, reference buttons, and command buttons. Guide does not distinguish between the author and the reader (Brown, 1987).

Intermedia (Garrett, *et al.*, 1986; Yankelovich, *et al.*, 1988) was the system developed at the Institute for Research in Information and Scholarship at Brown University. A "web" is a collection of nodes and links. A document is represented as a node. A link is a link in HAM. The web is defined by attaching an attribute named "web" to each link. A link filter is applied by using the predicate "web = mail", which will returned only nodes that are part of mail. A block is a piece of a document to a link which is anchored and can be any part in the application. The attributes provided by HAM allow the flexibility to efficiently model these relationships.

NoteCards (Halasz, 1987; Halasz, *et al.*, 1987) was a general-purpose idea-processing hypertext system developed at Xerox PARC. Its basic framework is a semantic network composed of notecards connected by typed links. It is a collaborative idea processing system, supported by a hypermedia system, for designers, authors and researchers to analyse information, construct models, add annotations to other nodes, formulate arguments, and process ideas. The system consists of four basic constructs: NoteCards, links, browsers, and FileBoxes. NoteCards contains information embedded in text, graphics, images or other media. Links represent binary relationships between cards and FileBoxes. FileBoxes can be represented in the HAM using nodes, links and attributes. FileBoxes provides a mechanism to organise cards into topics or categories. Both FileBox and NoteCard are equivalent to nodes. Browsers display node-link diagrams of portions of the network. The attribute is used to determine whether a node is a FileBox or a NoteCard. Link attributes specify which links refer to other FileBoxes and NoteCards. Nodes can reside more than one FileBox (Halasz, 1987; Halasz, *et al.*, 1987).

2.7.2 AHAM

AHAM (Adaptive Hypermedia Application Model) (Wu, *et al.*, 1998) is a model for adaptive hypermedia application, especially for educational applications, based on the DEXTER (Halasz and Schwartz, 1994), a formal reference model for hypertext systems. The web can be mapped onto the DEXTER model. Dexter realised the potential of holding links externally, or non-embedded, in some form of link database. Therefore, the link anchor became a first-class concept. In order to extend the DEXTER model to the field of AH, the AHAM model – which is identical to the Dexter model – uses similar within component, anchoring and runtime layers. However, the AHAM model focuses on the storage layer, the anchoring and the presentation specifications. The storage layer is divided into four parts: a domain model, a user model, a teaching model and an adaptive engine. A component's information consists of a set of attribute-value pairs, a sequence of anchors and a presentation specification. A domain model uses concept components for an abstract representation of information to describe application domain structure at both conceptual level and the level of information fragments and pages. All concepts are uniquely addressed through the anchoring layer. In addition, the domain model has sets of tuples that describe the relation between concepts. A user model represents the user's knowledge of the concepts in the application domain. It is an overlay model in that each concept that the user knows will be stored in the user model. Other user model information, for example background knowledge or preferences can also be stored in AHAM by storing this as additional attributes. A teaching model or adaptation model describes pedagogical rules as a guide to paths in the application domain. A teaching model is represented as a set of event-driven adaptive rules, defined in tuples. The variables will be changed based on the user's actions. The last component, an adaptive engine is the software environment used for adapting content and links based on the user model (Wu, *et al.*, 1998). The DEXTER model is illustrated in Figure 2-3 and the AHAM model is illustrated in Figure 2-4.

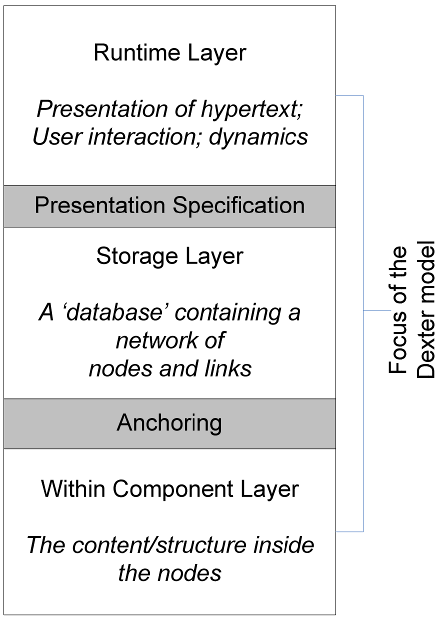


Figure 2-3 The Dexter Model (Halasz and Schwartz, 1994)

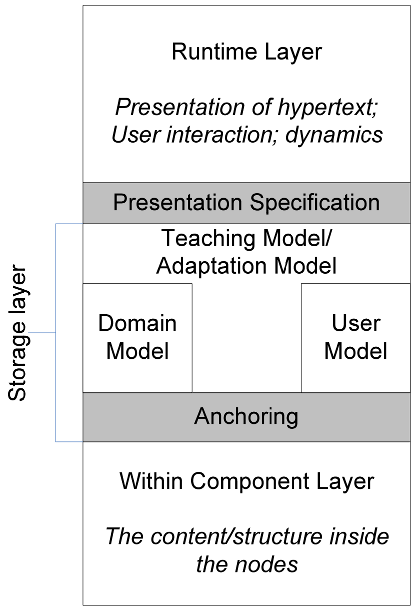


Figure 2-4 The AHAM Model (Wu, et al., 1998)

It is not only AHAM model that based on Dexter, but also the Munich Reference Model (Koch and Wirsing, 2002). The Munich Reference Model is a basis for a software engineering approach to support adaptive hypermedia developed by non AH-experts. It is an extension of the Dexter model, describing the network of nodes and links and the navigation mechanism, by adding a user model and adaptation model as part of the storage layer. The dynamic acquisition of user behaviour, a dynamic rule-based adaptation and a user behaviour trigger the Run-Time session.

The Munich Reference Model is different from AHAM in that the AHAM specifies the adaptation rule language, while the Munich Reference Model uses an object-oriented specification. The Munich Reference Model is described with Unified Modelling Language (UML) which provides the notation and the object-oriented modelling techniques (Koch and Wirsing, 2002). Figure 2-5 shows the Munich Reference Model.

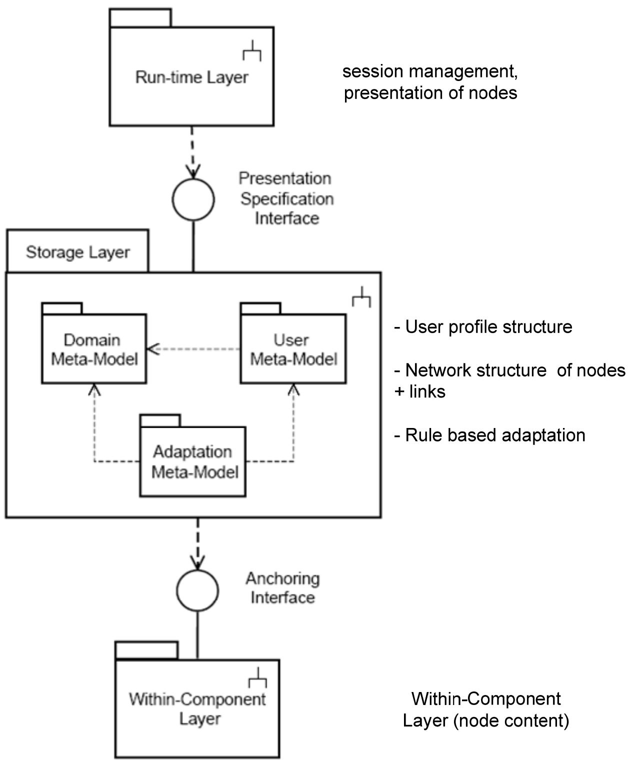


Figure 2-5 The Munich Reference Model (Koch and Wirsing, 2002)

2.7.3 HA^{3}L

HA^{3}L (Hypermedia Adaptation Using Agents and Auld Linky) is a server-side, agent-based AH application called Agent-Based framework for Adaptive Hypermedia (ABAH) (Bailey, 2002).

HA^{3}L system is the integration of an agent-based framework, SoFAR (Southampton Framework Architecture), with link service technology (described in Section 3.6), and Auld Linky (described in Section 3.6.1). HA^{3}L uses Auld Linky to demonstrate the use of FOHM model (described in Section 3.5) in supporting adaptive hypermedia techniques over the domain of "Rheumatology" for medical students. HA^{3}L system architecture consists of three agents: user model, interface, and adaptation agent. The user model agent keeps and manipulates records of a user's interaction with the system. The interface agent, like the interface component, communicates between the user and the adaptation agents. The adaptation agent provides processing which is required to adapt content and links based on a user's requirements. There are three information models inside adaptation agent: domain data file, domain structure (concepts, link, relationships) and adaptation rules (rules base containing the rules for applying adaptation). The reason that the domain structure is stored separately is because it supports data reuse and enhances interoperation of systems (Wiil and Whitehead, 1997). The ABAH framework is shown in Figure 2-6.

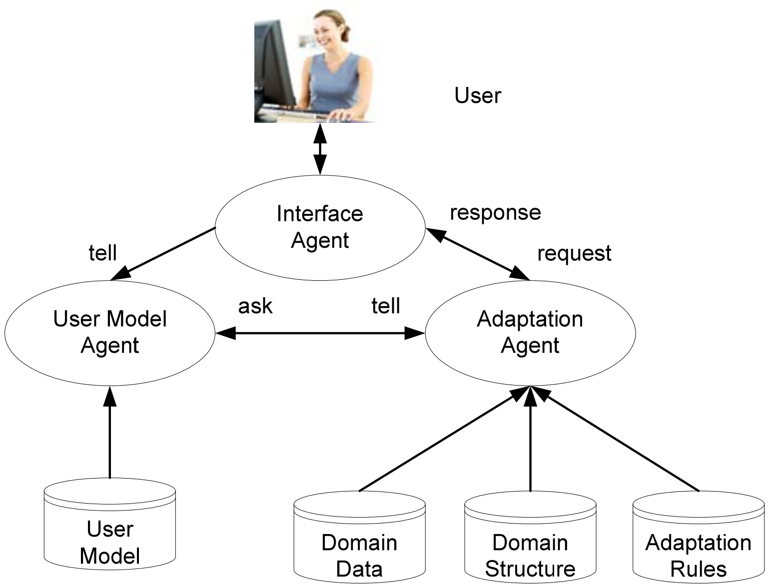


Figure 2-6 The basic ABAH framework (Bailey, 2002)

Bailey stated that using the ABAH framework, it provides flexibility in that it supports client-side, server-side and hybrid systems. It also enables agents to respond to queries from outside the ABAH framework, allowing not only data but services to be used by adaptive and non-adaptive applications. Moreover, from research on the ABAH framework, there arose another definition "*an adaptive web-based system is any web-*

based system that stores a model of the user's preferences and combines this with a set of adaptation rules to provide tailored services to clients" (Bailey, 2002).

2.7.4 Unified Component-based Architecture for Adaptive Hypermedia

A unifying component-based architecture for adaptive hypermedia was proposed by Ohene-Djan (Ohene-Djan, *et al.*, 2003). Figure 2-7 shows the architecture. In this generic architecture for an adaptive hypermedia system, there are four basic components.

- *Hypermedia structures* can be a single or flat formal text, or a hyperpage.
- *Metadata aims* to describe or *annotate* the content and behaviour of a hyperpage.
- *User model* is an important *component* for adaptation, stores individual users' information goals and history.
- *Engine component* provides a personalisation mechanism and an inference mechanism.

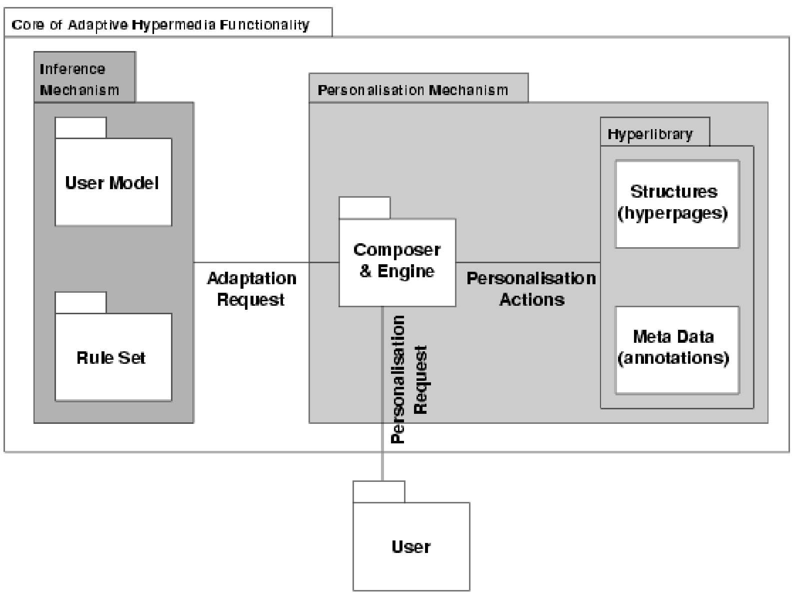


Figure 2-7 An Architecture for Adaptive Hypermedia (Ohene-Djan, *et al.*, 2003)

2.7.5 Generic Adaptation Framework for Web-based Hypermedia Systems

Paramythis and Stephanidis proposed a generic framework for development of adaptive web-based hypermedia systems and services (Paramythis and Stephanidis, 2005). Adaptation in this framework means the ability to capture and represent knowledge suitable for any object: for example different users, contexts and purposes. The main characteristics of this framework are: support for declarative specification of adaptive system behaviour; composition of adaptive hypermedia techniques from lower-level adaptation actions; inherent support for different approaches to representing and evaluating user model and context model and also adaptation logic; domain-independence, coupled with provisions for capturing the semantics and specifications of individual application domains; and lastly, applicability to any *document-centric* hypermedia system with *XML-based document representation* (Paramythis and Stephanidis, 2005).

The process of adapting documents involves the interoperation of the *decision-making component*, the *adaptation engine* and *modelling components*. The decision-making component decides which adaptations are to be performed. The adaptation engine applies adaptation decisions, expressed through adaptation actions. Then, adaptation decisions require access to the adaptation models which can be the user model, the context model, the domain or application model, for example. The adaptation models are encapsulated by the modelling components. Adaptation actions are performed as lower-level building blocks which can be used in isolation or in combination to provide higher-level adaptation techniques, both adaptive navigation and adaptive presentation. A high-level view of the framework components and their interactions is shown in Figure 2-8.

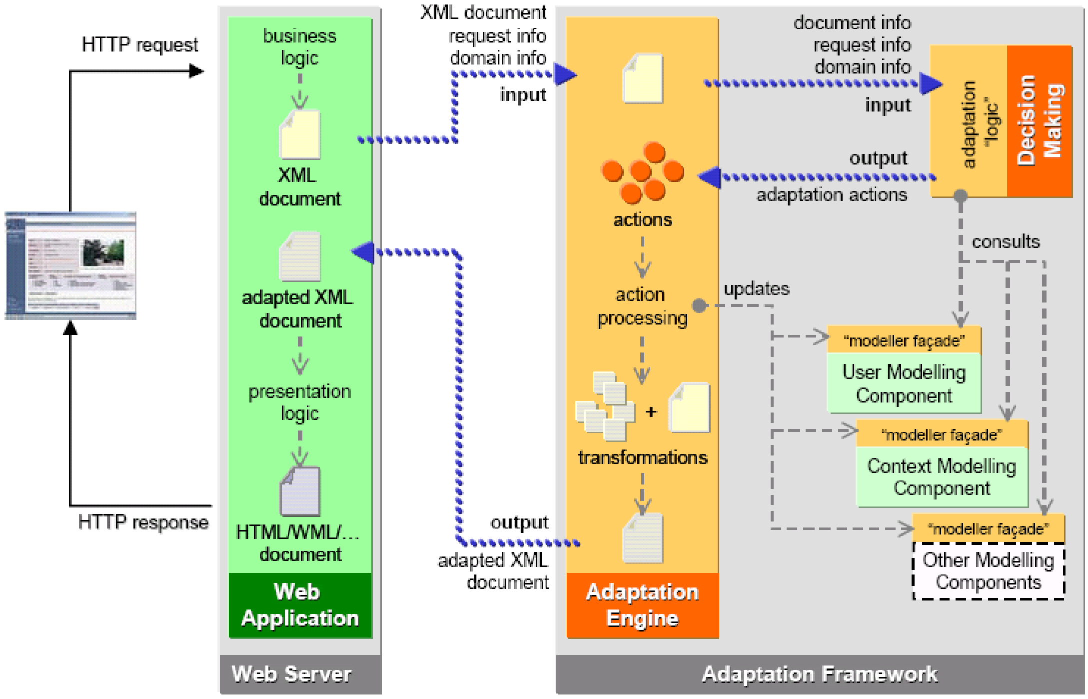


Figure 2-8 A high level components and interactions of framework (Paramythis and Stephanidis, 2005)

This framework, illustrated in Figure 2-8, has been applied in PALIO project, a European Commission funded research project that addressed the issue of universal access to community-wide services, based on content level adaptation and interface level adaptation, beyond desktop access. The PALIO system is implemented on top of the Cocoon [1]publishing framework.

2.7.6 Hunter Gatherer

Hunter Gatherer (schraefel, *et al.*, 2002) is a (browser-based) tool, a Firefox Extension [2] , which emphasised the user interface for capturing chunks of web pages constantly. It enables users to collect information chunks from within web pages, view those elements in a collection and edit those element collections. To collect chunks, Hunter Gatherer adopted a "reference-based approach" which automatically captures the location of the current page where data are captured, thus enabling users to trace back to the source easily. To edit the collection, users are allowed to rename the chunks in collections. The chunks in collections are dynamic in that only the references to locations, not the files, are stored. Therefore, chunks in the collection will be the most

[1] https://addons.mozilla.org/en-US/firefox/
[2] http://cocoon.apache.org/

current version. It is claimed that this strategy is similar to the open hypermedia concept of creating collections of *smaller-than-page* size elements for what Garzotto refers to as *pick-up* style (Garzotto, *et al.*, 1994), which is a popular method for building arbitrary collections. The collections, which came from referencing locations within documents are similar to user-defined composites of anchors that are constructed *by reference* rather than *by value* (Halasz, 1987). By referencing components with a URL rather than by copying the content, it applied Nelson's transclusions (Nelson, 1995; Nelson), in order to allow authors to control both private and public organisation. In order to avoid the *bumping* effect, Hunter Gatherer adapted Phelps and Wilensky's Robust Intra-document Location algorithms (Phelps and Wilensky, 2000) for reattaching annotations to altered components to keep track of *bumped* components.

2.8 Discussion

All models presented in the previous section refer to a domain model, which contains concepts and their relationships in the applications, while models for adaptive hypermedia (not the HAM model and Hunter Gatherer) include a user model for capturing a user's information. The user model is a core component for the adaptation process. Although the specific implementations of each model are different, they are conceptually identical.

It is claimed that a component-based framework or a modular framework will be a future direction of adaptive hypermedia systems (Brusilovsky, 2001) (De Bra, *et al.*, 2004). History shows that application development is moving in a modular direction, from low-level through high-level languages, component programming, object-oriented programming, multi-agent systems and now to a service oriented architecture. With a modular or component based architecture, various system components can be developed separately to perform their goals and interact with each other through a mediator. It increases platform independence and reusability of components, and increase the interoperability of systems. Lowe and Hall mention that during hypermedia development, it is advantageous to reuse hypermedia elements since it improves productivity, quality, reliability and maintainability. In order to reuse information effectively, information representation should be independent of the application representation (Lowe and Hall, 1999). This development methodology has been used in adaptive hypermedia frameworks.

There has been not much research into the area of interoperability, shareability and reusability within adaptive hypermedia systems. However, other fields have studied. Open hypermedia has studied the effects of interoperability, while Semantic Web attempts to provide mechanisms to give more definition and meaning, as metadata, to resources in order to support interoperability, shareability and reusability among various applications. Hunter Gatherer is a system which proposes to provide information sharing.

The adaptive hypermedia principle has been adopted in the research. According to the research motivation, given in Section 1.2, the framework has to provide an adaptable web-based system that delivers annotations based on users' profiles. Therefore, the framework should have a user model for keeping users' preferences, a domain model, an adaptation model and an adaptation engine. The framework should be in a modular framework in order to support shareability and reusability.

2.9 Summary

This chapter reviewed what hypermedia and adaptive hypermedia are. The history, terminology and technologies in this field were also presented. Adaptive hypermedia techniques can be classified to adaptive presentation and adaptive navigation. Adaptive presentation techniques are adaptive multimedia, adaptive text presentation and adaptation of modality, while there are six methods for adaptive navigation: direct guidance, adaptive link sorting, adaptive link hiding, adaptive link annotation, adaptive link generation and map adaptation.

The HAM is a model for hypertext while AHAM and Munich Reference Model are example models for adaptive hypermedia systems. Frameworks from adaptive hypermedia systems are also presented, HA^3L and a generic adaptation framework. All of these models can be unified as a component-based architecture for adaptive hypermedia which consists of four components: user model, composer and engine, hyper structures and metadata.

Chapter 3 Open Hypermedia

3.1 Introduction

This chapter presents the field of Open Hypermedia (OH), concepts, philosophy and models that have influenced this thesis. The evolution of open hypermedia research, from the hypertext community through the development of link services, and protocols for open hypermedia systems, including contextual link server are presented.

3.2 Overview

Most of the early hypermedia systems, were closed systems. In closed systems the type of files were bound to the applications and the file needed to be converted if it was to be used in other systems. It was difficult to extend the basic link model, which reduced flexibility of the system. In addition, links were embedded within the documents with particular format for the applications. For instance, the World Wide Web has anchors that are embedded within the HTML, i.e. <A HREF= “file.html”>source anchor</A> (Weal, 2000). In other words, in closed system data or links are not allowed to be accessed from outside applications without converting them to be used in another system as the linking mechanism is integrated tightly into the applications (Davis, *et al.*, 1992; Hall, 2000; Hothi, 2000).

In the 1990’s, open hypermedia research originated from the idea that hypermedia functionality should be made accessible across applications (Davis, *et al.*, 1992). Open hypermedia defines the provision of a link service, which enables client applications to manipulate links freely, creating, editing, and activating links (Fountain, *et al.*, 1990). The underlying principle of this approach is that links are considered as first-class objects. These entities are manipulated separately from hypermedia documents and stored independently in link databases (*linkbases*). Links are then added to hypermedia documents by means of a *link service*. The advantages of the link service approach are that links can be created, added, and edited without affecting the original document. By moving hyperlinks out of documents and into link databases, the relationships between documents are separated from the document content. Therefore the collection of documents becomes more maintainable, quicker to produce and easier to reuse. If there is any change of target documents, this only requires a change in the affected linkbases. Therefore using external linkbases enables different sets of links over the same content for different audiences and tasks (Carr, *et al.*, 1998). According to (Hall, 1994b) open hypermedia systems allow the dynamic generation of links causing less necessary of buttons. In an open hypermedia philosophy, hypermedia links are considered as a valuable store of knowledge. This knowledge can not be applied to any other data if it is bound to particular documents. In order to support this statement, only information about links would be stored in linkbases, while original data would be stored in native file format (Fountain, *et al.*, 1990; Wills, *et al.*, 2004).

3.3 Example of Open Hypermedia Systems

This section describes some of the early open hypermedia systems.

3.3.1 Intermedia

Intermedia (Yankelovich, *et al.*, 1988) was developed at Brown University's Institute for Research in Information and Scholarship. It was an environment designed to support authors, browsers and developers. It made no distinction between the roles of readers. The system integrated five applications for various purposes: a text editor (InterText), a graphics editor (InterDraw), a scanned image viewer (InterPrix), a three-dimensional object viewer (InterSpect), and a timeline editor (InterVal). The hypermedia functionality was integrated into each application while all applications shared a common set of links maintained independently of each application. Users were allowed to impose different sets of links on the same document. The system supported bidirectional links; a link could be followed in both directions from a particular section in one document to a particular section in another document. Intermedia allowed users to create anchor points, called *blocks*, in the documents. The block size could range from a whole document to a point. In Intermedia, links were maintained independently by a link server. Intermedia supported a multiple linkbase (Bailey, 2002). The limitation in the open hypermedia system aspect of Intermedia was that it had a closed link service so that the links could not be accessed or manipulated by the outside environment (Yankelovich, *et al.*, 1988).

3.3.2 Microcosm

Microcosm (Fountain, *et al.*, 1990) was an open hypermedia system for use in teaching and research. A link service was first developed, in 1989, at the University of Southampton as the heart of the Microcosm system. It was an open model for the hypermedia system. According to its open architecture, Microcosm enabled documents to be associated with other third party applications without any effect to the documents (Davis, *et al.*, 1992). There were several principle criteria in the design of Microcosm. Users were able to author their own hyperlinks across various types of document formats and applications. These links were stored separately from documents and maintainable in database of links (linkbase). In addition, these links were able to apply to data in other applications (Carr, *et al.*, 1994; Hall, *et al.*, 1995).

The linking mechanism was enhanced in 1992 (Carr, *et al.*, 1994; Davis, *et al.*, 1992) to support more sets of link types. *Specific links* pointed from an object to a specific point in the source document. *Local links* pointed from an object to any point in the source document, and *generic links* originated from any part at any point in any document. Both local and generic links were able to be applied to other application because of dynamic source anchors.

The essential components in Microcosm were viewers, filters, and the document management system which used an even-driven message-based interface. A viewer was any program that allowed users to view documents in any format. When the user clicked on a hyperlink or chose to follow links, the filter manager sent and received messages from the viewers to the Link Dispatcher which would examine the messages before offering to the user the actions to be performed. The document management system (DMS) managed data about documents, including attributes and pointers, about documents. However, the filters were independent processes. The important filters were linkbases containing the information about links in the system, including source and destination information (Carr, *et al.*, 1994).

Microcosm was based on modular design approach. Therefore, it was easy to extend. The links could be created dynamically according to different algorithms (Carr, *et al.*, 1994).

3.3.3 Sun's Link Service

Sun's Link Service (Pearl, 1991) was the first practical implementation of a link service, which was intended for developers to use as a toolkit to add hypertext functionality into existing applications on Sun workstations in a distributed environment. Sun's Link Service comprised a link database service which managed both shared and private collections of links. The link service functioned as a central mediator to provide link functionalities for other applications, which registered with the link service, through a well-defined protocol. Similar to Intermedia, Sun's Link Service also offered bidirectional links across applications (Bailey, 2002).

3.3.4 Chimera

Chimera (Anderson, *et al.*, 1994; Anderson, *et al.*, 2000) was a client-server open hypermedia system which provided hypermedia services in the heterogeneous software development environments. It allowed software developers to create links between software objects using various types of applications. The principal data model consisted of elements: objects, viewers, views, anchors and links. *Objects* were named and persistent entities. *Viewers* were named active entities, presenting objects which were specific to the viewer and the type of object. *Views* represented a pair of viewer and object (viewer, object); one object could be displayed by more than one viewer (multiple views). Anchors were managed by viewers according to a view. A link was a set of anchors related to views. In Chimera, links were first-class objects and were not embedded in a data object, unlike HTML links (W3C). Chimera's links were equivalent to Microcosm's specific links, while local links could be modelled using whole-component anchors (Halasz and Schwartz, 1994). In this environment, objects can have one or more views.

3.4 Open Hypermedia Protocol: OHP, OHP-NAV, FOHM

A protocol is required in order to achieve inter-operation between Open Hypermedia System (Millard, *et al.*, 2000). Hypermedia systems can be classified into three domains:

- *Navigational:* Navigational hypermedia is the most widely used, for example in the World Wide Web, and Microcosm. Navigational hypermedia allows users to click on the links created by authors, and to move between parts of the documents or within a document.

- *Spatial:* Spatial hypermedia enables users to organise information visually. It allows users to manipulate the information by visual characteristics, such as colour, size, or closeness of their relationships with existing nodes.

- *Taxonomic:* Taxonomic hypermedia is the categorization of information into hierarchies.

There are several protocols developed for open hypermedia as the following.

- The Open Hypermedia Protocol (OHP) (Davis, *et al.*, 1999) was the first protocol developed by the Open Hypermedia Systems Working Group (OHSWG).

- OHP-Nav was developed as a text-based protocol and concerned with the navigational field.

- The Fundamental Open Hypermedia Model (FOHM) is based on the OHP model, with additional context-awareness features (Michaelides, *et al.*, 2001).

3.5 Fundamental Open Hypermedia Model (FOHM)

The Fundamental Open Hypermedia Model (FOHM) (Millard, *et al.*, 2000) is a protocol for representing the structural elements of open hypermedia systems with additional context-awareness features. It is a data model for expressing hyperstructure and it does this by representing associations between source and destination nodes.

3.5.1 FOHM Objects

The four essential components of a FOHM structure are the Data object, Association, Reference and Binding.

- *Data objects:* are wrappers for any piece of data that lies outside of the scope of the FOHM model. In the open hypermedia view, data can be of any format, such as text, audio, video, etc.

- *Associations* are structures that represent relationships between Data objects.

- *References objects* are used to point at Data objects or at parts of Data objects.

- *Bindings* specify the attributes of the connection between Association and Data objects.

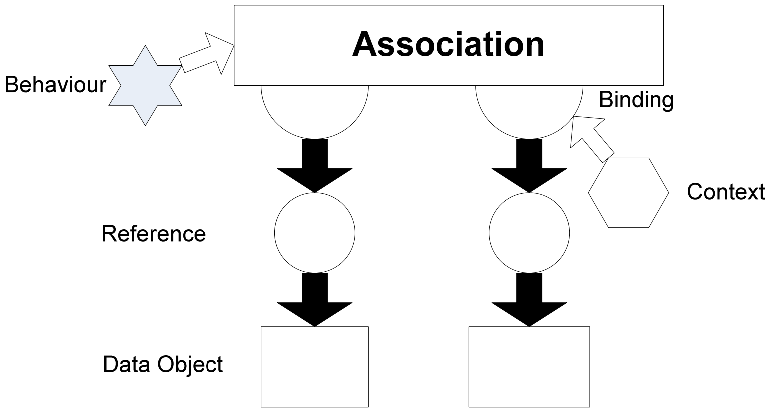


Figure 3-1 A FOHM Model

FOHM also provides two modifier objects which can be attached to any part of the FOHM structure: these are Behaviour and Context objects. The behaviour objects are used by client applications, whereas the context objects define conditions for the visibility of particular objects to individual users. In other words, the behaviour objects are used to inform client applications that dealing the FOHM structures about which actions to be performed at a given event condition. For example, the behaviour object is used to modify the user model when the user visits a particular data object in the structure. The context objects specify which part of the structure can be seen. For example, the data object with context object of ‘technical’ that has the value ‘true’ will be selected as illustrated in Figure 3-6. A FOHM model is shown in Figure 3-1.

3.5.2 FOHM Structure

FOHM structures can be used to support several adaptive techniques according to Brusilovsky's taxonomy (Figure 2-1) (Bailey, 2002). These structures, Navigational link, Tour, Level of Detail and Concept, are presented as the following.

- *Navigational Link* is an association assigned with link typed endpoints: source, destination, or bi-directional. FOHM also supports *regions* which allow start or destination points to refer to a specific location of media type in FOHM data objects. Figure 3-2 contains a single source (SRC) and two destinations (DEST) data objects. The source location references a particular segment in the object, while the destinations are pointed to text and image objects. There are also context objects attached to the FOHM structure. This is fundamental and used in the thesis.

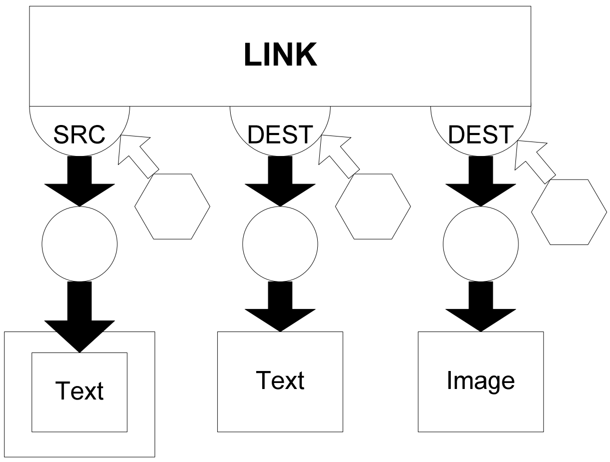


Figure 3-2 A FOHM Navigational Link

- *Tour* is an association representing an ordered set of objects. The objects in a tour can be data items or other associations. An example in Figure 3-3 shows that there are three objects in position1, 2 and 3. The data item in position 3 is pointed to a Link association.

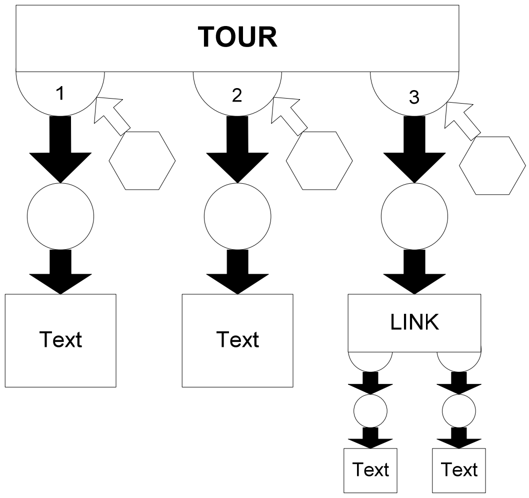


Figure 3-3 A FOHM Tour Structure

- *Level of Detail*: is an association representing an ordered set of objects with similar conceptual information but in greater detail or complexity. For instance, a Level of Detail (LoD) can be used to represent a concept where the first object is a summary, the second object is an introduction and the third object contains a full description of concept with examples, as illustrated in Figure 3-4.

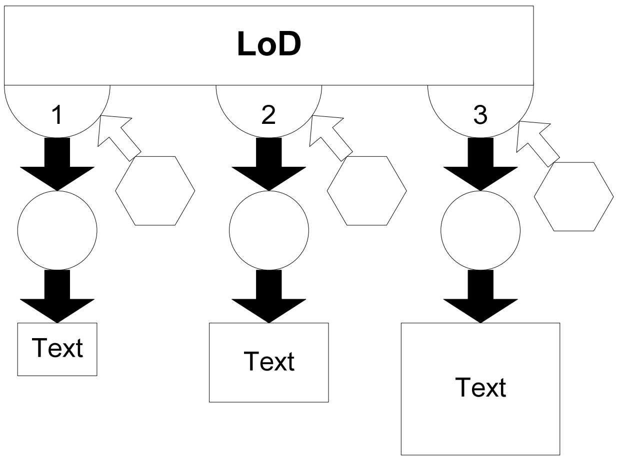


Figure 3-4 A FOHM Level of Detail Structure

- *Concept*: is an association representing a collection of data items with similar conceptual information but they can be in different media types. In this case context objects are used to specify the media type. An example is shown in Figure 3-5.

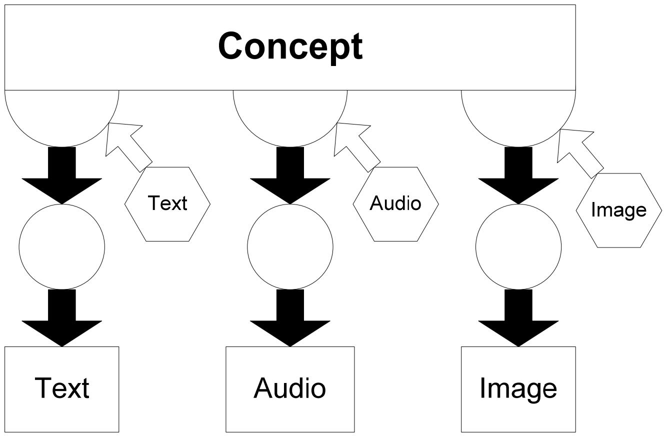


Figure 3-5 A FOHM Concept Structure

3.6 Link Service Approach

The link service approach originated from the open hypermedia community. This approach considers links as core objects and links are stored independently, and separated from document content, in external link databases (linkbases). Therefore, a linkbase can be considered as a database of link structure. There were several advantages of using linkbases (Ciancarini, *et al.*, 2002) (Carr, *et al.*, 1996; Carr, *et al.*, 1998; Carr, *et al.*, 2002; Davis, *et al.*, 1992; Hall, 2000; Hothi, 2000). Different link sets can be provided on the same set of documents for different readers and activities. Links are allowed to be created on media that they have no rights to modify. Intensional (specific) links and generic links (links associated with a string instead of a position) could be added to the documents. Moreover, linkbases can be verified and updated centrally if there is change, thereby providing more control, enhancing the maintenance and reusability of documents than embedded links.

How to provide links to documents has been an interesting research issue. For example, Bailey investigated on combining the concept of open hypermedia with adaptive hypermedia methods and techniques (Bailey, *et al.*, 2001). According to the research, link augmentation technique was defined as "*a technique whereby external links are inserted directly into the body of a document*" (Bailey, *et al.*, 2001). Link augmentation tried to add more useful or related information as links to the existing documents, while link annotation was the process concerning visible properties of hyperlinks, such as the link's text, style or appearance (Bailey, 2002).

Context can be used in order to offer relevant links to the documents. El-Beltagy presented a method, by applying context analysis, for generating links adaptively based on individual user's needs, helping users with their navigation and information finding. There were several factors that could affect the context of users, for instance: user's interests, level of expertise in various topics, the user's role in an organisation or their physical location. Contexts of a document could be captured in various ways, such as its contents, its format (i.e. HTML, gif), its purpose or the date it was created. The model defined two factors as context; interests of user and contents of document within which the links are to be rendered. This concept was implemented as a multi-agent system in the QuIC project (El-Beltagy, *et al.*, 2001).

3.6.1 Auld Linky as Contextual Link Server

Auld Linky (originally called Auld Leaky) (Michaelides, *et al.*, 2001) is a Contextual Open Hypermedia Link Server and it is the latest generation of the link service. Michaelides, *et al* (2001) used FOHM (described earlier in 3.5), which provides an underlying data model, to design Auld Linky. Auld Linky stores FOHM structures in XML objects and provides the mechanisms (APIs) for storing, looking up and querying FOHM structures by matching, termed context culling process, by removing the part of the structure that the context modifier fails to match and sending the resulting parts of the structure that matches the users' requests back to the client applications. Auld Linky is designed to provide the light weight link service to be installed and executed operations over simple HTTP connections; GET, POST, PUT, DELETE, based on XML.

For example, in the POST operation, when requests (in a FOHM structure as XML message) for querying the linkbase are sent to Auld Linky, Auld Linky will transform it into an internal FOHM structure. The FOHM structure will then be pattern-matched against each FOHM structure in linkbases. A set of matches will be returned back to the clients as XML message (Michaelides, *et al.*, 2001).

An example of how Auld Linky works with a simple FOHM Navigation Link structure is shown in Figure 3-6. This is a link with one source (the location with the word "FOHM") and two destinations (with URLs). Both destinations explain "FOHM", the first with no technical (Tech=False) information while the second is with technical detail (Tech=True). If the structure was loaded into Auld Linky and queried using this context then Auld Linky would remove the inappropriate destination.

Several adaptive hypermedia applications can be modelled with the FOHM structures and features. The Context object is used by Auld Linky, a context link based server, in order to distinguish between which bindings should be returned (Michaelides, *et al.*, 2001). It means that part of the structure that did not match with the context on the structure was removed before it was given back. Behaviour objects can be used by client application for updating the user model by providing keywords. These keywords can then be used to query FOHM subsequently by attaching context modifiers. The process is shown in Figure 3-7.

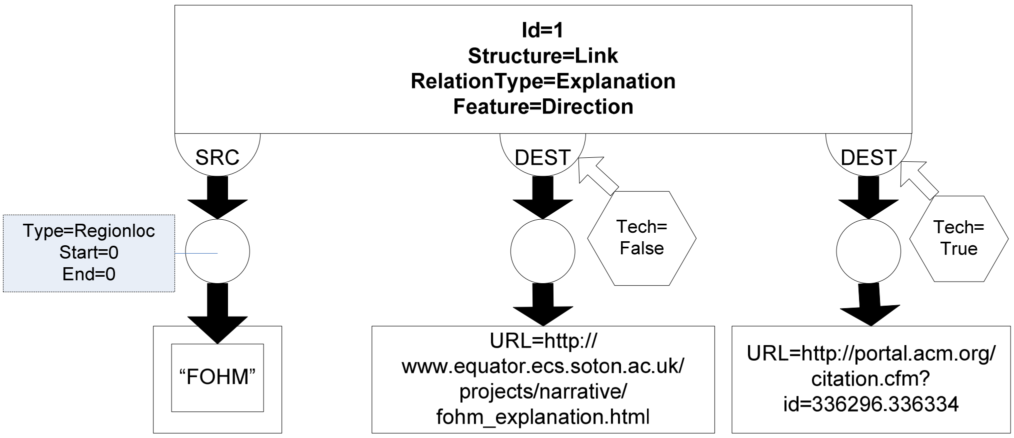


Figure 3-6 A simple FOHM Navigation Link

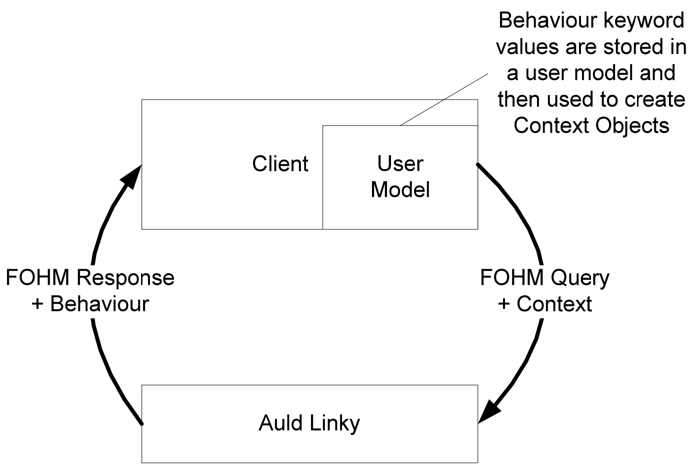


Figure 3-7 The process of User Modelling in Auld Linky (Bailey, 2002)

3.6.2 Distributed Link Services (DLS)

A Distributed Link Service (DLS) was implemented for use with Web-based systems and was based on the open hypermedia principles of Microcosm (Davis, *et al.*, 1994).

The link server facilities were implemented as pseudo modules stored in the link databases. They provided functions for creating, editing, and the traversal of links. Users could configure these linkbases into several categories according to different contexts based on a user's requirements. The link databases recorded the source and destination attributes of the link and the type of the link (Carr, *et al.*, 1998). The date and time the link was created and a link description were also recorded to solve the link integrity problem (Davis, 1995).

The 'interfaceless' approach, discussed by Carr, was an attempt to make the DLS transparent to the user. This approach allowed the links to be embedded in the document, by compiling a chosen set of links into a specific set of documents, and then sent to the browser by a particular adapted World-Wide-Web proxy server. However, there were some disadvantages to this approach. It changed the 'reader-led navigation' to 'click on a predefined choice'(Hall, 1994b). In addition, the link compilation was sent through the WWW protocol only, thereby limiting its application. Also, the links could be applied to any document but the document formats that it was able to support were limited (Carr, *et al.*, 1996). There have been researched on integrating the DLS with the Semantic Web technologies to provide the linking based on concepts, which will be presented in Chapter 4.

There was an attempt to provide a storage mechanism for private, shareable links on any HTML or XML document on the Internet, called the XLinkProxy (Ciancarini, *et al.*, 2002). The XLinkProxy was a proxy-based external linkbase management system using XLink and XPointer. It was concluded that it was worthwhile to separate content from presentation by using external links, in that, private links, alternative link sets or generic links could be provided to documents. However, there were a few limitations in the implementation, for badly-behaving HTML or XML documents.

Carr said about the comparison between DLS with XLink in his lecture notes that DLS provided explicit (positional) addressing, implicit (content-based) linking and DLS had knowledge as it recognises about data objects (such as people or citations), meaning of contents or presentation (i.e. formatting), while XLink, on the other hand, did not. XLink needed to be tied in with the XML processing model, so the position had to be specified in processing (Carr, 2007).

3.7 Related Research

FOHM has been used to implement to provide several adaptive hypermedia systems. There was an attempt to integrate Adaptive Hypermedia Architecture (AHA!), an Adaptive Hypermedia system developed by the Eindhoven University of Technology, and Auld Linky using the ideas from Sculptural Hypertext (Bernstein, 1998; Michaelides, *et al.*, 2001). It aimed to discover how the structural approach affects the processes of adaptation within AHA!, and whether the OH system could support a more predetermined adaptive model. However, they found that the implementation of combination of two systems was not completely successful because of the very different methodologies of freeform hypertexts and deterministic hypertexts. Therefore, some of their fundamental design principles were lost during integration of two systems (Millard, *et al.*, 2003).

Abdullah proposed an adaptive environment around SCORM to supply learners with learning materials suitable for teachers and students. The work provides a real-time personalisation service for SCORM using FOHM and Auld Linky. The system uses the concept and user's preferred learning style, stored in a user model, to query the link server. Then, at run time the link server return supplementary links according to the given concept name and user's preferred learning style (Abdullah, 2006).

3.8 Summary

This chapter presented background of fundamental research to the open hypermedia work underpinning this thesis. Open hypermedia systems enable client applications to manipulate links freely by separating links from documents, storing them in a linkbase, and treat them as first-class objects. This approach offers advantages such as reducing authoring and maintenance effort, applying links to various documents, and enhancing reusability of documents.

Some open hypermedia systems are presented, Intermedia, Microcosm, Sun's Link Service and Chimera. Protocols for increasing interoperability between open hypermedia systems, including models form expressing the domains of hypertext have been researched. FOHM is a protocol for interoperation between open hypermedia systems with additional context-awareness features. FOHM structures: Navigational Link, Tour, Level of Detail and Concept, can be used to support many adaptive techniques.

A link service approach allowed links to be manipulated separately from documents, therefore it is possible to describe resources and store the information with links referring to them in linkbases. The Auld Linky is a contextual link server which used the FOHM model as a data model. The Distributed Link Service (DLS) provides functions for creating, editing and traversal of links for Web-based systems. The DLS proxy is an attempt to make the DLS transparent to the user through a particular Web proxy server. However, there are some disadvantages to this approach as it limits the application, and changes the 'reader-led navigation' to 'click on a predefined choice'.

While there have been researched on providing adaptive hypermedia techniques in open hypermedia field, open hypermedia has been extended to cover issues such as reasoning mechanism on metadata, reusability, shareability and integrating open hypermedia with the Semantic Web technologies.

According to requirements in Section 1.2, the open hypermedia concept is used to separate annotations from document contents and the annotations are managed independently through an annotation service (a.k.a link service in this chapter). FOHM could be extended to represent hyperstructure and supports contextual annotation which is required from Section 1.2. However, the adaptive hypermedia, presented in Chapter 2, and open hypermedia do not completely fulfil the requirements to enable users to add contexts to an annotation or view annotations based on their personal profiles. Therefore, the Semantic Web idea is also applied.

The next chapter will present the primary research area of the work in this thesis, the Semantic Web which is the cutting edge of the Web including advantages of adding semantic to resources in the Web.

Chapter 4 Semantic Web

4.1 Introduction

This chapter presents the field of the Semantic Web which is another area that influenced this thesis. The concepts, frameworks and applications in the Semantic Web will be presented.

4.2 Overview

HTML web displays information on the web, by using simple links. However, HTML web provides less support for the meaning of links, automations and search engines. Metadata can be used for linking resources on the web, but defining web resources with conceptual content-based information enables search engines to find more relevant information (Carr, *et al.*, 2001). The Semantic Web (Berners-Lee, *et al.*, 2001), coined by Sir Tim Berners-Lee, is a new generation of the web. It is an effort to provide a common framework allowing resources to be shared and reused, interoperated and extensible across application, enterprise and community boundaries. The Semantic Web is an attempt to build a new World Wide Web architecture that annotates content, as metadata, with formal meaning in order to make web content more accessible and understandable (Noy and McGuinness, 2001). The Semantic Web content is more structured than current HTML Web. The Semantic Web develops an environment in which human and software agents can communicate on a semantic basis. In addition, the Semantic Web provides a machine-understandable document in that it indicates a machine's ability to solve well-defined problems – resource discovery, information brokering, and information filtering – by performing well-defined operations on existing well-defined data. Rather than asking machines to understand natural language, it involves asking people to produce well-defined documents (Bechhofer and Goble, 2001; Berners-Lee, 1998; Berners-Lee, *et al.*, 2001) (Herman, 2001). The definition of Semantic Web is illustrated in Figure 4-1. Semantic Web can be used for various kinds of purposes by enabling the machines do more work, for example, aggregation of syndicated content, improved searching, bookmarking, annotation, and collaborative tool operations. Various applications can be connected by concepts

(Berners-Lee, 2003). Gardner claims that Semantic Web technology can significantly improve the effectiveness of digital resource sharing (Gardner, 2007).

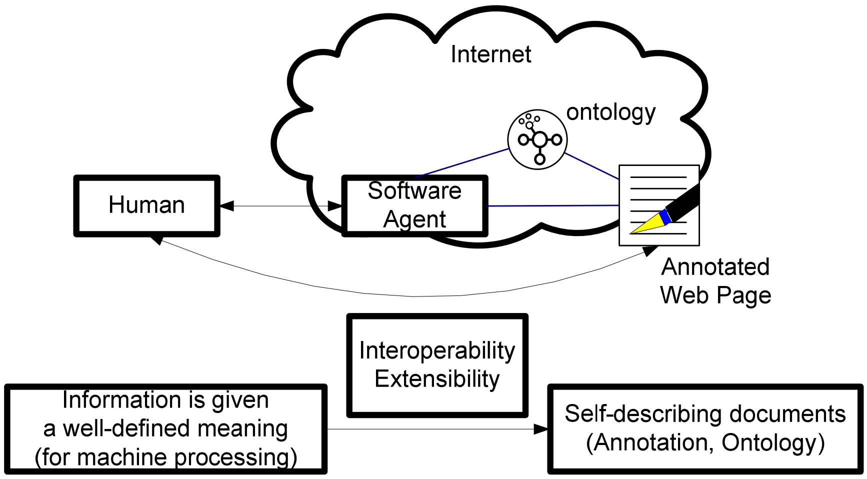


Figure 4-1 Semantic Web definition

In February 2004, The World Wide Web Consortium released the Resource Description Framework (RDF) and the OWL Web Ontology Language (OWL) as W3C Recommendations. RDF is used to represent information and to exchange knowledge on the web. RDF schema provides a simple ontology language. OWL is used to publish and share sets of terms (ontologies), supporting advanced web searching, software agents and knowledge management. The layers of the Semantic Web architecture will be presented in the next section.

4.3 The Semantic Web Layer Cake

To express meaning in the Semantic Web, there are several layers, namely URIs and Unicode, XML, RDF model and schema, ontology, rules, logic, proof, trust, signature and encryption. Figure 4-2 represents the Semantic Web layer. On the bottom layer, all objects or resources on the web are referred to or identified by URIs (Uniform Resource Identifier) and document encoding method (Unicode[3]). The other layers provide the metadata and knowledge aspects of the Semantic Web.

[3] Unicode is a system for the interchange, processing, and display of texts from diverse languages

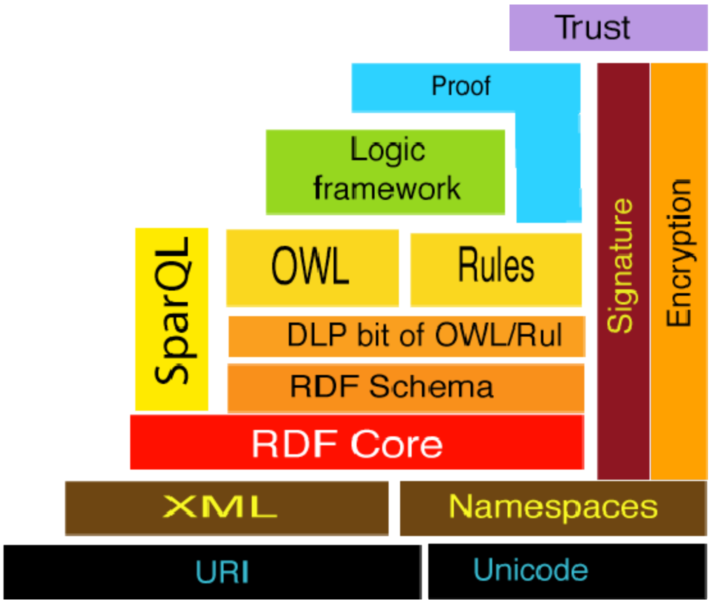


Figure 4-2 Layers of the Semantic Web (Berners-Lee, 2005)

The schema layers, XML and namespaces, RDF model and syntax and RDF Schema, provide structural and modelling capabilities. XML (W3C, 2006) provides a surface syntax to structure documents but without semantic constraints for meaning of documents. The Semantic Web assumes that there will be several and, perhaps, overlapping vocabularies. Therefore, XML namespaces allow the construction of different XML vocabularies by providing a means of uniquely identifying every item in every vocabulary. An XML document consists of a properly nested set of open and closed tags. Each tag can have attribute-value pairs. The most interesting point of XML is that tags, and the way that these tags are constructed are not predefined. For this reason, applications have to determine these tags themselves. XML Schema (W3C, 2004b) is a language for controlling the structure of XML documents, while XML Query language (W3C, 2007) provides query facilities in standardised way through semi-structured data stored as XML.

The next schema layer, Resource Description Framework (RDF) (W3C, 2004c), model and syntax, is a data model, based on XML, for describing any resources as metadata or attributes of these. Resources and properties in RDF are identified by URIs. The idea of using URIs to identify things promises a global and unique naming scheme. RDF defines a simple data model of triples: subject, predicate, object. An RDF statement is an object-attribute or property-value triple. An RDF statement can be represented as a labelled, directed graph. An arc of the graph represents a property while a node represents any "thing" or object. Properties describe relations between resources or objects. An example of a statement represented in a graph is shown in Figure 4-3. In RDF terms, this graph states that "the resource "http://www.wealth-society.com" (a URI) has the property "authored by" with the value "Thanyalak Maneewatthana". The RDF triple is also extensible, meaning that one object of the statement can become a subject others. For example, in Figure 4-4 the object http://users.ecs.soton.ac.uk/~tm03r/ becomes the subject of two other statements. In this way, semantics can be added to any resources supporting machine processing. In addition, RDF enables users to share information resources easily and also provides interoperability of applications to exchange machine-understandable information on the web. However, there is no built-in semantics or vocabularies in RDF. Therefore, the next layer, the RDF Schema (W3C, 2004a) is a vocabulary for describing classes and properties and other resources. With the RDF Schema, relationships between classes can be defined. But the triple is not enough. An ontology language can describe or constrain concepts and the relationships between them.

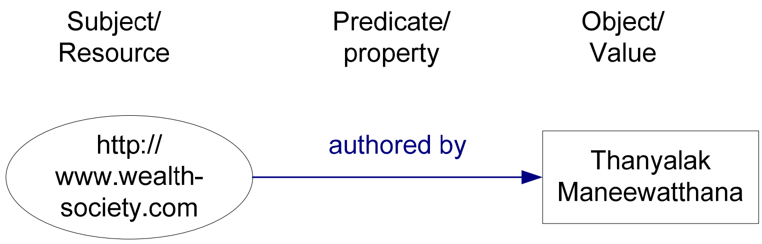


Figure 4-3 a RDF triple: describing a resource in a graph

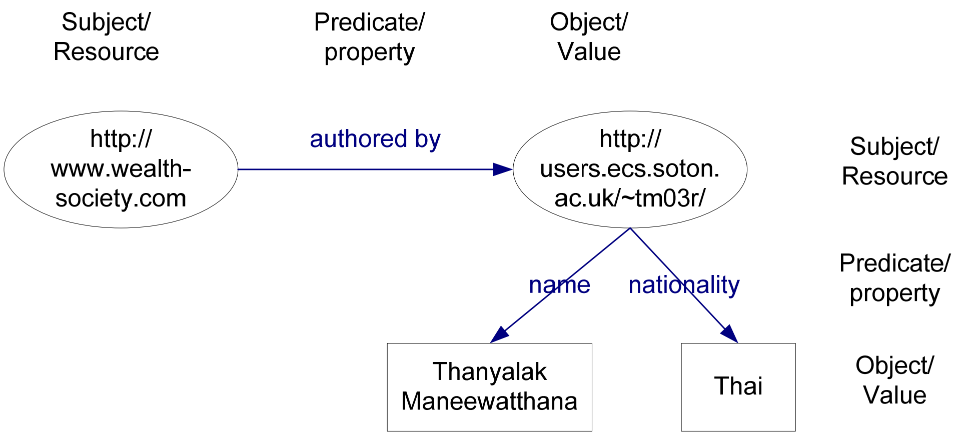


Figure 4-4 An extension to the RDF triple

The ontology layer supplies more meta-information to define concepts and relationships precisely using constructs which are not available in the schema layers. Ontology is "*a specification of a conceptualisation*" [Gruber, 1993]. It represents the formal, agreed, common understanding about semantics and relationships of information objects. To define shared and common knowledge in domain of interest, an ontology can make documents or information both human and machine communicable, and supports the exchange of syntax and semantics (Bechhofer, *et al.*, 2001; Gruber, 1993; Guarino, 1998; Stojanivic, *et al.*, 2001). Ontologies enable software agents to find the meaning of content within web resources (Carr, *et al.*, 2002). Moreover, ontologies facilitate knowledge sharing and reuse between agents (Davies, *et al.*, 2003b). The ontology layer has now been instantiated with the Web Ontology Language, OWL (Bechhofer, *et al.*, 2004).

The logic, proof, trust, signature and encryption layers are the remaining Semantic Web architecture. The logic layer adds inference ability to the Semantic Web. For example, it enables the derivation of new data from existing information or checking of a document against a set of rules for self-consistency. The proof layer enables processes to assure that a statement is true by using a series of inferences. The trust layer provides processes with the ability to guarantee resources and the statements that they contain. In other words, the trust layer provides a mechanism to establish the validity of statements. The digital signature makes it possible to use public key cryptography to secure a document (Berners-Lee, 1998).

Figure 4-5 represents how the layers of the Semantic Web, interoperate together.

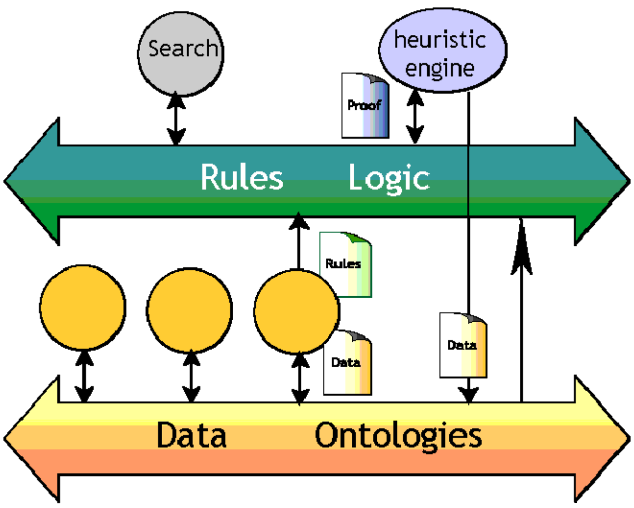


Figure 4-5 Semantic Web bus (Berners-Lee, 2000)

In the Semantic Web concept, knowledge can be represented and derived in the Semantic Web applications.

4.4 Knowledge on the Semantic Web

There are several hierarchies of content or information proposed. According to Ackoff the content of the human mind can be categorised as the follows (Ackoff, 1967).

- *Data*: data are raw symbols that have no significance beyond their existence, no meaning of itself, for example numbers ('2'), text ('Sam'), or images.

- *Information*: information is data that has been given some useful value by way of relational connection. It can provide answers to "who", "what", "when", "where" questions (e.g. the creator is Sam).

- *Knowledge*: knowledge is what is learned throughout time from data and information, indicating a greater understanding of some area (e.g. 'a creator, such as Sam, is interested in the topic: knowledge management'). It is a deterministic process. Knowledge comes from memorised information, but does not provide any further knowledge.

- *Understanding*: understanding is an analysis, and then synthesise of new knowledge from previously held information, knowledge and understanding. The difference is that understanding comes from learning while knowledge comes from memorising.

- *Wisdom*: wisdom is the evaluation of understanding to produce a new understanding. It is non-deterministic, non-probabilistic process. It is stated that computers do not have and never will have the ability to possess wisdom (Bellinger, *et al.*, 2004).

While Ackoff proposed the hierarchy as data, information, knowledge, understanding and wisdom, Bellinger stated that understanding is not a separate level of its own, but it can support the transition from each stage to the next, as illustrated in Figure 4-6 (Bellinger, *et al.*, 2004).

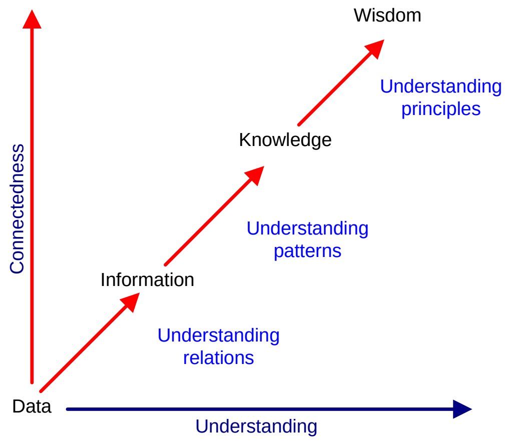


Figure 4-6 Diagram of Data Information Knowledge Wisdom transition (Bellinger, *et al.*, 2004)

There are several approaches to structure the data to create knowledge for machines to read and understand. For example, Kampa described the structure in an order of increasing semantic detail as list, vocabulary, thesaurus, taxonomy and ontology (Kampa, 2002).

- *List*: an enumeration of words (e.g. shopping list).
- *Controlled vocabulary:* an enumeration of terms within the same domain that have been listed explicitly (e.g. technical terms in computing).
- *Thesaurus:* an enumeration of terms within a domain. It describes the standard terms for concepts in a controlled vocabulary. It uses associative relationships such as synonyms and more complex relationships, such as broader or narrower terms, related terms and other forms of words (e.g. medical terms dealing with surgery), metaphors.
- *Taxonomy*: taxonomy is a collection of controlled vocabulary terms organised into a hierarchical structure. Each term is in one or more parent-child relationships (subclass, superclass) to other terms in the taxonomy.
- *Ontology*: a taxonomic structure with constraint mechanisms and relationships between terms.

An ontology is therefore a formal model of representing knowledge that provides a common agreement of concepts and objects existing in the real world together with the relationships between them. It provides a common vocabulary for users who want to share information in the domain. Ontologies enable not only knowledge sharing and reuse, but also improved communication between people and software entities. In addition, ontologies are used to separate domain knowledge from operational knowledge (Chandrasekaran, *et al.*, 1999). It is claimed that ontologies have proven highly successful in many disciplines, such as multi-agent systems (Falasconi, *et al.*, 1996) and bioinformatics (Stevens, *et al.*, 2000), (Sidhu, *et al.*, 2005).

Berners-Lee claims that if the Semantic Web is designed properly, it can help the evolution of human knowledge. The Semantic Web enables anyone to publish every concept with a URI and these concepts can be linked worldwide. This structure supports knowledge construction, meaning that humans can develop meaningful analysis of software agents which enable humans to work and learn together (Berners-Lee, *et al.*, 2001). Noy and McGuinness state that there is no one way to model a domain correctly. The ontology development is an iterative process, and existing ontologies should be reused (Noy and McGuinness, 2001).

A key contribution by the Semantic Web is to provide a set of worldwide standards (Uren, *et al.*, 2005). The Semantic Web technologies, especially ontology, offer an opportunity to improve knowledge management capabilities in large organisations (Maedche, *et al.*, 2003). Ontologies enable knowledge modelling, knowledge inference techniques and formalised conceptualisation of domains (O'Hara, *et al.*, 2002). It has been demonstrated that Ontology-based knowledge management, such as On-to-Knowledge (Fensel, 2002), can improve information access in the large intranets of organisations, facilitate electronic knowledge sharing and reuse for customer relationship management, and promote knowledge management in virtual organisations. Kampa implemented the E-Scholar Knowledge Inference Model (Kampa, 2002) to demonstrate how ontologically-modelled data can be used to infer new facts and resolve links based on an analytical question. Benjamins stated that in knowledge management, ontology-based retrieval is preferable to keywords-based retrieval since ontology-based retrieval allows accessing implicit knowledge and presenting information collected from distributive locations in a coherent way to users, because there is knowledge of how the retrieved information relates to each other (Benjamins, *et al.*, 1998).

4.5 Annotation for Knowledge Management

Annotation is important to the context-building process (Miles-Board, 2004), influence on readers (O'Hara, *et al.*, 1998) and scholarly communication (Furuta and Urbina, 2002). Marshall points out that annotation can be constructed in several ways: as link making, as path building, as commentary, as marking in or around existing text, as a de-centering of authority, as a record of reading and interpretation, or as community memory (Marshall, 1998). The dimensions of annotations suggested by Marshall can be: formal or informal; tacit or explicit, the functions of annotation from a reader's point of view and the roles of annotations as they are used to communicate with others (public, private or share, who the audience is). According to Ovsiannikov users create annotations to remember the main aspects of a document which reminds easily with access through the annotation (Ovsiannikov, *et al.*, 1999). Critical remarks, questions, notes and ideas reflect personal opinion and support readers in thinking about the document content. Devising content in personal words helps in clarifying certain aspects of a paper, and the sharing of annotations is important in a collaborative environment. User can see what is important from annotations that have been made in a community. Annotations support understanding, memorisation and later retrieval (Schilit, *et al.*, 1998) (Vasudevan and Palmer, 1999). Annotation is an important part of *active reading* (Adler and Van Doren, 1972).

Active reading is the combination of reading and critical thinking and learning (Adler and Van Doren, 1972). It is a fundamental part of educational and knowledge work which involves not only reading, but also underlining, highlighting and scribbling comments (Adler and Van Doren, 1972).

The following section will briefly describe the annotation systems for the web, and annotation on the Semantic Web will be presented thereafter.

4.5.1 Annotating the Web

Marshall mentioned that annotation has contributed to foundational work in the Hypertext field (Marshall, 1998). For example, Bush's Memex (Bush, 1945) focused on annotation through trail blazing, Engelbart's NLS (Engelbart, 1963) concerned a capacity for a journal system commentary and Nelson's Xanadu introduced a transclusive approach (Nelson, 1999). Several systems have been proposed to annotate documents on the World Wide Web for example ComMentor, Yawas and CritLink.

ComMentor is a browser-based (NSCA Mosaic [4]) tool as part of the Stanford Integrated Digital Library Project (Röscheisen and Mogensen, 1995). The architecture is based on *annotation sets*, meaning that every annotation belongs to a particular set and annotates a particular page at some specific location. Every set is associated with a particular server, the *annotation server*, and identity, like a URL. Each annotation set can be assigned access control: public, private or workgroup. ComMentor supports a thread-like discussion based on an idea in the original document. It means that annotations are documents themselves, therefore readers can annotate annotations. To decrease the problems that comments disappear from a user's view, ComMentor provides users with *landmark*, *tour* and *trail* facilities. The landmark allows users to set reference points. The tour guides users through a list of all annotations. Trails can be applied to implement multiple guided tours through the same content without confusing the users. This means that trail marks are added to each set which form a linked tour according to the semantics of the set and allow users to select which tour to follow. ComMentor can provide trail facility because metadata-information is managed independently of the documents on separate meta-information servers. In other words contents and connecting superstructures are separated and dynamically synthesised together based on a chosen user context (Röscheisen, *et al.*, 1995). The research prototype was completed in 1994, and the code of the tool is no longer maintained (Staab, *et al.*, 2001).

Yawas is an annotation tool that is based on the Document Object Model (DOM) and Dynamic HTML (Denoue and Vignollet, 2000). It codes the annotations into an extended URL format and uses local files similar to bookmark files to store and retrieve the annotations. A modified browser can then transform the URL format into DOM events. Locally stored annotation files can be sent to other users (Staab, *et al.*, 2001).

CritLink is a web-based annotation system based on the proxy approach (Yee, 2002). Annotations are stored and saved in an annotation server at http://crit.org through the CritLink Mediator. The system is simply used by prefixing the URL with http://crit.org. For example, to see the annotated version of semanticweb.org someone can access the

[4] http://archive.ncsa.uiuc.edu/mosaic.html

system with the URL http://crit.org/http://semanticweb.org. The mediator then loads the page and adds in any annotations stored in the annotation server. Annotations can be added at any point in the document. The selected text to be annotated must be retyped [sic] into the commenting window. It is noted that CritLink supports public annotation and critical discussion because CritLink provides bi-directional links, extrinsic links, fine-grained anchor and typed links (the definitions of each are described in Section 2.3). Bi-directional links enable users to follow backwards pointers and therefore an annotation can be a web document with a link to the annotated document. Extrinsic links allow others to annotate a document without requiring co-operation from the author. A fine-grained target anchor enables the annotation to refer to a particular phrase in the document. Finally, typed links allow the annotation to specify a relationship to the target document.

Since the Semantic Web emerged, there have been attempts to apply the Semantic Web to annotation. According to Uren, the benefits of applying the Semantic Web to annotation are enhancing information retrieval and improving interoperability (Uren, *et al.*, 2005). Information retrieval is improved by searching capability, meaning that the systems can make inferences about data. The interoperability for heterogeneous resources can be achieved by annotating these based on a common ontology. Human annotators are prone to error. There are several systems to support the mark-up of documents and support knowledge management. The following are examples of annotation and applications in the Semantic Web.

4.5.2 Annotation Frameworks and Systems in the Semantic Web

Annotea

The World Wide Web Consortium's Annotea project [5] aims to provide a mechanism for users to add and publish shared inline annotations to web documents (Kahan, *et al.*, 2001). The definition for the term *annotation* is that given by the W3C is any object (*has a destination anchor*) such as notes, comments or ideas, that are attached (*has a source anchor*) to a web document or selected part of the web document (Koivunen, 2001). Annotea is based on an infrastructure which combines existing open W3C RDF with XPointer, XLink and HTTP. Annotations in Annotea are modelled as a class of

[5] http://www.w3.org/2001/Annotea/

metadata and they are viewed as statements about a web document. Annotations are stored separately from the documents in one or more *annotation servers* which are generic RDF databases. Annotea-enabled browsers, Amaya editor/browser [6] and Annozilla [7], can manipulate annotation servers and query them to obtain annotations for pages.

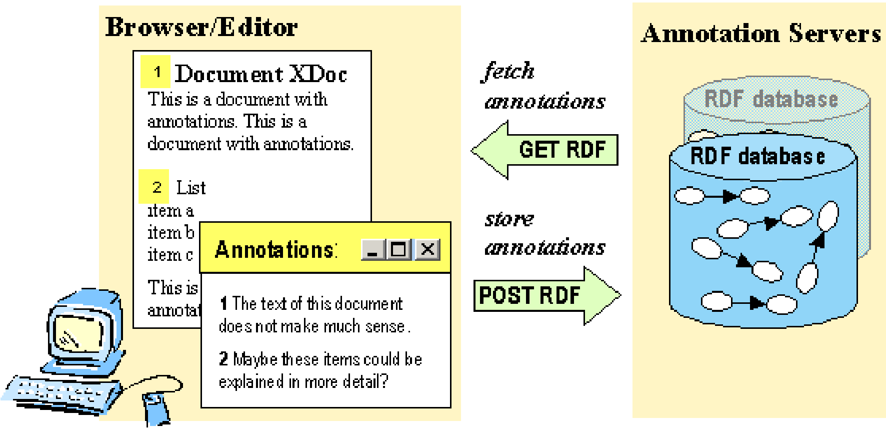


Figure 4-7 Basic architecture of Annotea (Kahan, *et al.*, 2001)

In Annotea, annotations are described in RDF schema and are stored in annotation servers. Because annotations are first-class Web resources each annotation is associated with a URI. Users can query a server to retrieve an existing annotation, post a new annotation, modify an annotation or delete an annotation. These operations between a client and an annotation server are accessible through HTTP. Figure 4-7 shows the basic architecture of Annotea.

Annotation statements consist of the following main parts: the body of the annotation containing the content of the annotation, the link to the annotated document with a location within the document, an identification of the person making the annotation and additional metadata related to the annotation. The type of an annotation is defined by the users or can be created as additional classes by them. The general annotation super-class is called *Annotation* [8]. Other basic annotation classes are subclasses of *Annotation* class. Basic classes are *Advice*, *Change*, *Comment*, *Example*, *Explanation*, *Question* and *SeeAlso*, but new properties can also be created. The annotation properties illustrated in RDF model is shown in Figure 4-8, and schema definitions for properties

[6] http://www.w3.org/Amaya/

[7] http://annozilla.mozdev.org/

are given in Table 4-1 . As annotations are typed, users are able to classify or organise them.

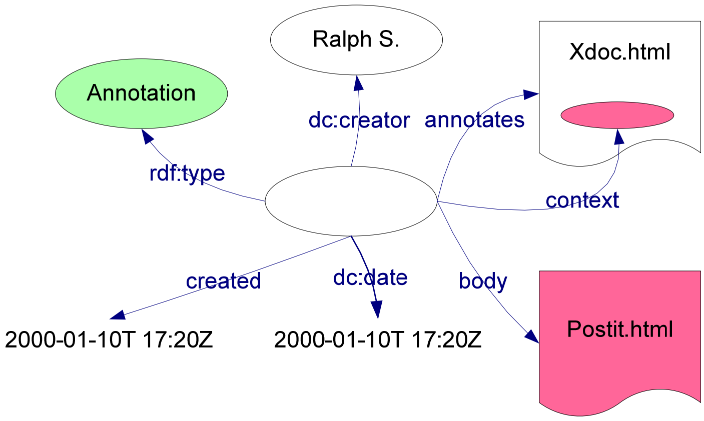


Figure 4-8 The RDF model of an annotation (Kahan, *et al.*, 2001)

rdf:type	The type of the annotation. It indicates the author's intention when making annotation. The value should be of rdf:type Annotation, or any of its subclasses.
annotates	The resource to which the annotation applies.
Body	The content of the annotation.
Context	The context defines where exactly inside the document the annotation is attached. It is an XPointer.
dc:creator	The creator of the annotation.
Created	The date and time on which the annotation was created.
dc:date	The date and time on which the annotation was last modified.
Related	Related resources that augment the *body* of the annotation.

Table 4-1 Basic schema definition of properties in Annotea (Kahan, *et al.*, 2001)

[8] http://www.w3.org/2000/10/annotation-ns#Annotation

Using an RDF schema type for modelling annotations allows developers to define new types of annotations. Koivunen demonstrates how Annotea can be extended to support *threaded annotation discussions* by adding a *reply-to* property that defines which annotation was the previous one in the thread, and a *root-of-thread* property that is the first annotation in the thread (Koivunen and Swick, 2001).

CREAM

CREAting Metadata (CREAM) is a framework for annotation environments that enables the construction of relational metadata using a *domain ontology* (Handschuh and Staab, 2002). It means that an entire document is defined as an instance of a particular concept in the ontology. The document is then annotated by dragging and dropping pieces of content into attribute in the ontology browser. These metadata are built based on a domain ontology. It uses ontology concept instances, considering evolving ontologies and offering annotation in a semi-automated way (De Bra, *et al.*, 2004). The first version of CREAM provided an annotation tool that implemented this framework, called Ont-O-Mat [9]. The CREAM framework and the tool have been extended to reduce the border between authoring and annotation. CREAM comprises inference services, crawler, document management system, *ontology guidance/fact browser*, *document editors/viewers*, and a *meta ontology*. Ont-O-Mat tool supports three types of interaction: annotation by typing a statement (involves working within the ontology guidance/fact browser), annotation by markup (reuse of data from the document editor/viewer in the ontology guidance/fact browser), annotation by authoring web pages (involves the reuse of data from the fact browser in the document editor). In CREAM, annotations are in RDF, while OWL and XPointers are used to locate annotations in text which restricts documents to web-native formats such as XML and HTML. CREAM proposes to solve problems concerned with consistency, proper reference, redundancy avoidance, relational metadata, ontology maintenance, and easy of use of tool, by combining advanced mechanisms for inference, fast crawling, document management and information extraction (Handschuh, *et al.*, 2001; Handschuh, *et al.*, 2002) (Handschuh and Staab, 2002).

[9] http://annotation.semanticweb.org/ontomat/index.html

AWF

Associative Writing Framework (AWF) is a framework for associative writing on the web. The work implemented the annotating web content, linking web content, visualising emerging hypertext, integrated writing in MS FrontPage, and publishing integrated hypertext. AWF applied open hypermedia, link integrity management, Semantic Web, hypertext writing and annotation approaches to provide an interface for supporting browsing, reading, annotation, linking and integrated writing. Annotations and links are stored separately from content by the AWF server. This AWF server is based on Annotea, the annotation server, an open structure service which provides annotation rather than links. RDF templates are used for representation and interchange. Annotations and links are managed externally, in order to allow different users to explore and contribute to a shared interpretation. Writers can create annotations as anchors and links on the Web pages. Link metadata; semantic type, creation time and creator are displayed as partially obscured links (*ghosted* connections) (Miles-Board, 2004).

SHOE

As noted by Bechhofer and Goble, SHOE (Heflin, *et al.*, 1999) and Ontobroker (Decker, *et al.*, 1999) concern semantic annotation using ontology. Ontobroker had originally been developed as a research prototype as part of the Semantic Web initiative. However, Ontobroker has matured and is therefore now a commercial product marketed as Ontoprise [10].

SHOE is a knowledge representation language that allows ontologies to be designed and used directly. In the SHOE approach, the ontologies are extended to documents by the SHOE HTML extension. The HTML documents are stored separately from the ontologies, which can be referred to using a URL. The power of SHOE is in supporting the use of multiple ontologies, while the similar formal knowledge representation, such as Semantic Web languages, is not provided. On the other hand, Ontobroker uses in-document markup, annotated by the <A> tag. Information retrieval in both SHOE and Ontobroker are processed by agents through a group of pages registered with a service

[10] http://www.ontoprise.de/content/e1171/e1231/index_eng.html

(Bechhofer and Goble, 2001). SHOE Knowledge Annotator [11] is a Java program that allows users to mark-up web pages with the SHOE ontology.

COHSE

The purpose of the Conceptual OHS Environment (COHSE) project was the integration of an open hypermedia architecture, especially the Distributed Link Service (DLS, described in Section 3.6.2) with ontological services, called *ontological linking*, to support an architecture for the Semantic Web and provide the linking based, on the concepts in the content that appear in web pages (Carr, *et al.*, 2001; Carr, *et al.*, 2002). The documents are linked according to the metadata annotated in the documents. With this approach documents are annotated based on description logic and augmented with the annotations dynamically at browse time or reading time. The ontological linking approach dealt with the *prolific linking* problem or too many generic links (Carr, *et al.*, 2001). The previous implementations of COHSE were a proxy and browser plug-in. With these implementations there is a limitation in supporting personalistaion techniques such as collaborative filtering (Wang, *et al.*, 2006). Since a portal provides a framework to aggregate content from different sources, and to provide a mechanism to store user profiles, customisation and also personalisation (Smith, *et al.*, 2004), COHSE has been implemented as a portal. COHSE has been extended to support customisation and to create an adaptable open system by using third party ontologies and services to discover resources on the web. COHSE supports content adaptation techniques, navigation adaptation techniques, presentation adaptation techniques, browsing history based adaptation techniques and collaborative filtering (Yesilada, *et al.*, 2006).

WiCK

Writing in the Context of Knowledge (WiCK) provides a way to apply Semantic Web technologies for an office environment for office workers. It focuses on creating and reusing documents enriched with knowledge, by modelling the knowledge flow in a business writing scenario such as writing a funding research project proposal. This approach aids authors in knowledge writing tasks by integrating a commercial office production environment with several knowledge web services. It produces information varied by user roles in the organisation, and by context and type of information. In

[11] http://www.cs.umd.edu/projects/plus/SHOE/KnowledgeAnnotator.html

addition, it provides a function for updating the knowledge bases with new knowledge when the proposal is completed. The Semantic Web is employed to attach; explicit statements of semantics to documents, knowledge about authors, and reasons for their publications. The Semantic Web is also used to infer contextually appropriate knowledge (Carr, *et al.*, 2004).

Magpie

Magpie is a tool supporting a *zero-cost* approach to Semantic Web browsing by automatically associating an ontology-based semantic layer with items found on a web resources (Dzbor, *et al.*, 2004). Magpie supports the *interpretation of web pages*. It provides the background information and knowledge relevant to a web resource for a user (Motta, *et al.*, 2003). The key principle of Magpie is that it enables a user to decide what semantic information is shown on their browser. With this tool, the interface enables ontological differences to be highlighted and the services provided are dependent on the class of entity found. Magpie is a semantic web browser and also a framework for developing semantic web applications. It means that Magpie enables users to navigate the web using both semantic and hypertext links and automatically associates semantics with items found on a web resource, and offers the relevant services or suitable functionalities to the user (Dzbor, *et al.*, 2004).

4.6 Adaptive Hypermedia in the Semantic Web

Semantic Web technologies have been applied to adaptive hypermedia systems (in general). As presented in Section 2.6, the user model, domain model, and adaptation model are key concepts in adaptive hypermedia systems. The Semantic Web provides mechanisms to identify resources to satisfy the requirements based on not only keywords but on knowledge, which increases the precision of the answer (Baldoni, *et al.*, 2005). For example, in educational adaptive hypermedia systems, ontologies and metadata standards for the web can be used to provide personalisation and reuse of content (Silva and Oliveira, 2004). This means that ontologies can be used to represent knowledge in the system, for example educational content, the domain to be taught, and the learner's profile. Ontologies are also used for interoperability to reuse of content.

In order to interact, exchange data and share components simultaneously with various adaptive hypermedia applications, De Bra *et al.* stated that an open and modularised architecture is the next step for adaptive web-based systems. Web services provide mechanisms for flexible composition of system components (as services) at different levels, such as conceptual, user model and adaptation. These different components communicate with each other through service invocations. The Semantic Web, especially the ontology, defines a system's terminology and the properties of each service, and also promotes the shareability and interoperability among the services (De Bra, *et al.*, 2004).

From the CHIME project (Chepegin, *et al.*, 2003), Aroyo presented a service-oriented framework, for adaptive web-based systems, to show how ontologies and the Semantic Web can improve the adaptation and the interoperability among adaptive hypermedia systems. The main components, domain model, application model, adaptation model and user model, are separated. The ontological approach is used to enable a shared understanding of concepts through the system and to provide semantic relationships between the information resources and the user's knowledge. The semantic descriptions enrich content, functionality and data flows, allowing the discovery, configuration and management of internal and external agents. Modularity of the architecture, the multi-agent architecture, and its openness are applied to interoperate with various adaptive applications or components. The shareability and reusability of modules is accomplished by using Semantic Web Services as this approach allows discovery, configuration and management of agents. Aroyo stated that by augmenting encapsulated system modules with rich formal descriptions the system management is improved. The architecture is shown in Figure 4-9.

Figure 4-9 illustrates the Semantic Web-based Adaptive Hypermedia Architecture. It is adapted from AHAM (Wu, *et al.*, 1998) and Munich (Koch and Wirsing, 2002) reference models by introducing the notion of services and semantic description of functionality in terms of ontologies. In this approach, four main system components are separated. Each system component is implemented as a web service and communicates through service invocations. The Domain Model Service is responsible for concepts of Domain Ontology. The User Model Service collects and analyses data about a user's activities. The Adaptation Model Service, an agent, plans and performs the adaptation. The Application Model Service contains a generic description of the user tasks. *Role-Goals-Tasks-Methods* chains (Problem Solving Methods) provides abstract descriptions of the reasoning processes which can be applied to solve tasks in a specific domain.

Bridges specify mappings between different model services within the CHIME framework. The last component, ontologies, defined by open standards such as XML or RDF, are used to define descriptions of each service, and to enabling sharing and interoperability in the context of the WWW. An advantage of the modularised architecture is that it provides a very high degree of flexibility (De Bra, *et al.*, 2004).

Henze proposed a modular framework, called the Personal Reader framework (PRF), for development and maintenance of personalised functionalities on the Semantic Web (Henze, 2005). PRF is a service-based architecture providing Visualisation Services for creating a user interface, and Personalisation Services for providing the personalisation functionality. The core component is the Connector Service which is the mediator between all Services in the PRF by passing user requests and delivering personal recommendations based on context (Henze, 2005). Figure 4-10 shows the architecture of PRF.

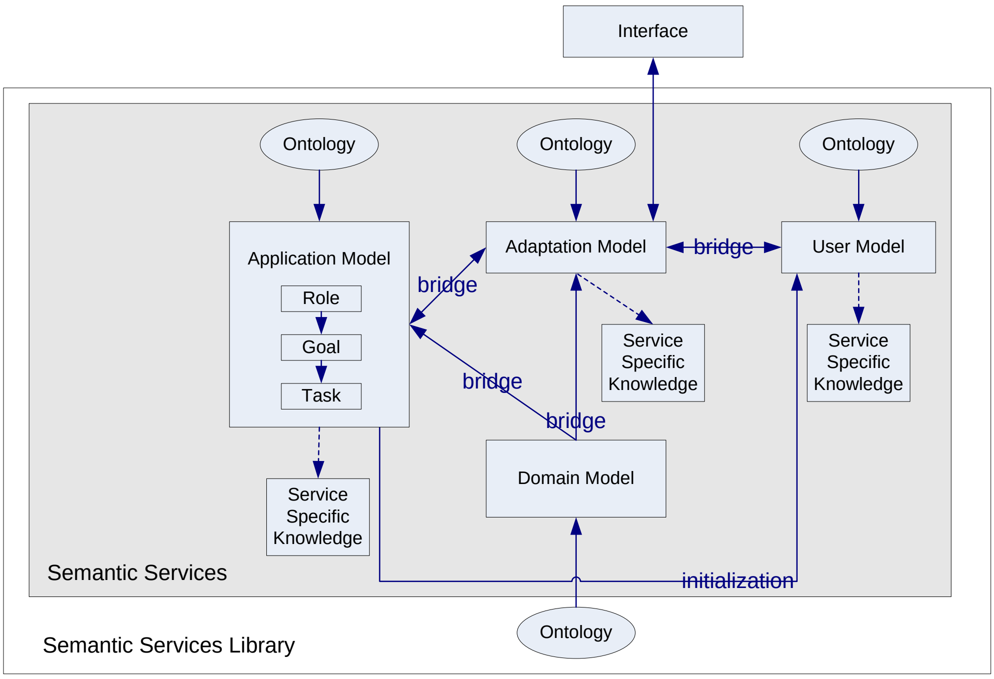


Figure 4-9 Semantic Web-based Adaptive Hypermedia Architecture (Aroyo, *et al.*, 2004)

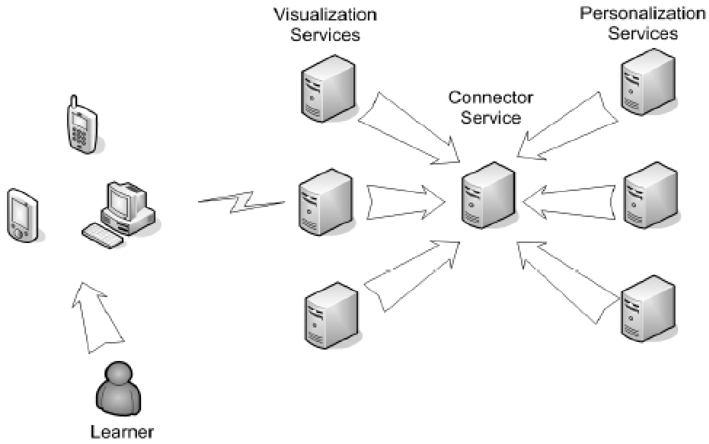


Figure 4-10 Architecture of the Personal Reader framework (Henze, 2005)

4.7 Open Hypermedia and the Semantic Web

According to Gibbins who investigated the relationships between FOHM (presented in Chapter 3.5) and the RDF(S)/OWL, FOHM and RDF are both models for expressing metadata, but they are concerned with different levels of structures. They did this by describing any FOHM structure in RDF and any RDF graph as FOHM association (Gibbins, *et al.*, 2003). While FOHM is a model capable of representing contextual structure for a variety of hypermedia domains, the RDF is also expressive enough for describing resources on the World Wide Web. FOHM is concerned with higher level generic structures while RDF functions at a lower level as a triple (subject, predicate, object). Implementing FOHM in RDF results in some semantic relationships being lost. The advantage of using RDF as the theoretical model can not be obtained. In addition, because applications are unable to identify the RDF graph as a FOHM structure, the semantics in FOHM is not provided.

However, if RDF is extended by using RDFS or OWL, the semantics of the FOHM structure can be described while the benefits of RDF still hold. Describing FOHM as an OWL/RDF ontology provides two types of component: the domain-neutral core ontology and the domain-specific ontologies. The domain-neutral core ontology defines basic notions of FOHM objects such as associations, bindings, and features. The domain-specific ontologies describe the different hypermedia domains, for instance navigational and spatial. The structure of FOHM objects is defined as the class model of Java implementation. Therefore, the semantic of relationships between the different classes of FOHM object is expressed explicitly.

For this reason, defining FOHM using RDF(S)/OWL provides several benefits. It is a consistent way of representing high level relations. Metadata can be attached arbitrarily with explicit predicates. There are RDF tools available, such as parsers, validators, and processors. In addition, it is possible to use RDFS/OWL to define the FOHM hypertext domains and extend definitions to explore new domains. Finally, it is possible to express formal ontology of structure and relation types: for example it could use a schema to define valid navigational link types for argumentative hypertext (Gibbins, *et al.*, 2003).

4.8 Discussion

Semantic Web technologies have been applied in adaptive hypermedia systems in order to provide more advantages; improved personalisation, interoperability, shareability and reusability. The service-oriented framework provides more functionality, especially interoperability with other applications.

The annotation definition used in this research is based on that used by the W3C which is the mechanism for users to mark and add information on the document (Kahan, *et al.*, 2001). According to the scenario given in Section 1.2, the systems should provide these features:

- manipulation of contextual annotations,
- personalisation of annotations by delivering annotations based on users' preferences,
- sharing of annotations
- reuse of annotations by adding other annotations into personal repository.

As different users might have different requirements, if the system displays all annotations available on the current web page then the information overload problem might occur, as described in Section in 1.2 and in Section 2.3. For this reason, adaptation of annotation is an important feature. Annotations made by users should be shareable with others and also be reuseable or transferable by adding other annotations into an individual repository. In addition, if annotations do not follow a standard then they will soon disappear, because the annotations can not be shared across applications.

Table 4-2 shows the properties of various annotation systems and frameworks in the Semantic Web presented in this chapter. Most of the tools or these frameworks do not provide personalised annotations, or reusability of annotations. ComMentor is a closed system. WiCK provides reusability of documents. However, there is no annotation tool or framework that can provide all the features required.

The definitions of properties in Table 4-2 are the following**.**

Annotation: ability to mark, manipulate and comment on information.

Shareability: the ability to allow other users to see annotations by setting authorisation for others to read each annotation.

Reusability: the ability to change some context of other peoples' annotations and save them to a personal space. The annotations are transferable.

Personalisation: the system ability to deliver annotations according to users' preferences.

Annotation Storage: the mechanism to store annotations: e.g. local / annotation server, RDF, XML file

Formats: formats of annotations, e.g. RDF/XLink, XPointer

Requirements / Systems	Annotation	Shareability	Personal repository (Reusability)	Annotation with context	Personalisation	Annotation Storage	Formats (Syntactic and Semantic Interoperability)
ComMentor	✓	Public, group, private	X	X	✓ (trail, tour)	Annotation server	X (Text)
Yawas	✓	Private (local file), public	X	X	X	Text file	X (HTML, DOM)
CritLink	✓	Public	X	X	X	Annotation server	X (HTML)
Annotea	✓	Public	X	X	X	Local or annotation server	✓ (RDF(S)XLink, XPointer)
CREAM	✓	N/A	X	X	X	N/A	✓ (RDF, OWL, XPointer)
AWF	✓	Private, share	X	X	X	AWF server	✓ (RDF)
SHOE	✓	N/A	X	X	X	Embedded in Web page	X (SHOE)
COHSE	✓	N/A	X	X	X	Annotation server, DLS	DAML+OIL
WiCK	✓	N/A	✓	X	X	Annotation server (3store)	X (MS smart documents)
Magpie	✓	N/A	X	X	✓	None, real-time	X (HTML OCML)

Table 4-2 Properties of various annotation systems

4.9 Summary

This chapter presented an overview of Semantic Web technologies and the related research relevant to this thesis. The Semantic Web provides a machine-understandable data in that it indicates a machine's ability to solve a well-defined problem, resource discovery, information brokering, and information filtering, by performing well-defined operations on existing well-defined data. Several layers of representational structure are used to express meaning in the Semantic Web; XML, RDF, ontology, rule, logic, proof, trust, signature and encryption. There are several systems that apply Semantic Web technologies to annotate the web, such as SHOE, COHSE, CREAM, Annotea, Magpie and WiCK. In addition, the chapter also presented research on applying Semantic Web to adaptive hypermedia and open hypermedia.

The next chapter will document the initial work and early investigations which provided the fundamental knowledge for the work in this thesis.

Chapter 5 Initial Work

5.1 Introduction

This chapter presents the experiments in the area of adaptive hypermedia, open hypermedia and the Semantic Web. The objective of conducting these experiments was to assess the limitations of a Link Service approach (described in Section 3.6) with the FOHM (presented in Section 3.5) together with the Semantic Web (described in Chapter 4) to support the sharing and reuse dynamic links according to a users' preferences.

The objective of the first experiment, the Semantic Web Learning Portal (SWLP), was to investigate the use of semantic web technology to implement an e-learning portal. The content of the web pages, in the portal, was augmented with dynamic links. These links were selected for inclusion based on the users' preferences. The links were implemented using FOHM structures and the content was implemented as RDF.

The second experiment, the Adaptive Personal Information Environment (*adPIE*), was used to investigate the possibility of providing shareability and reusability of information, and dynamic links, through the combination of a link service approach and Semantic Web technology.

These two experiments essentially provided the underpinning concepts for the framework, discussed in the next chapter which is the core of the thesis. This chapter concludes with how these two systems could support the research questions by showing the possibility to separate annotations for web-based systems and how to provide dynamic annotations, shareable and reusable annotations.

5.2 Semantic Web Learning Portal (SWLP)

The first experiment aimed to investigate the limitation of a link service, described in Section 3.6, and dynamically links when used to dynamically select links based on the user's context. In order to achieve this, an e-learning application was implemented, called the Semantic Web Learning Portal (SWLP). In the case of SWLP the context is the knowledge level a user has of a topic and thereby setting the criteria for which links are to be displayed on a webpage. In order to achieve this, links were represented in as FOHM structures and manipulated by Auld Linky (described in Section 3.6.1), the content as RDF and a SWLP proxy was developed as a server.

In order to support the research question, which is about separation of annotations from web pages, it was essential that links are not embedded in the web page but kept separated and are used to augmented the information in the web page. The learning material and users' information was defined in RDF. This allowed the application of semantic web technology to be used.

5.2.1 SWLP Proxy Server

The SWLP proxy server is a server that sits between a client application such as a web browser and a World Wide Web servers. The SWLP proxy serves links from specified linkbase(s) through an Auld Linky link server (described in Section 3.6.1). The Auld Linky link server parses FOHM structures that stores contents and structures of links held in linkbases. The links are inserted into the webpage before being sent to the client applications through the SWLP proxy server. The interaction between system environments with the SWLP proxy server is shown in Figure 5-1.

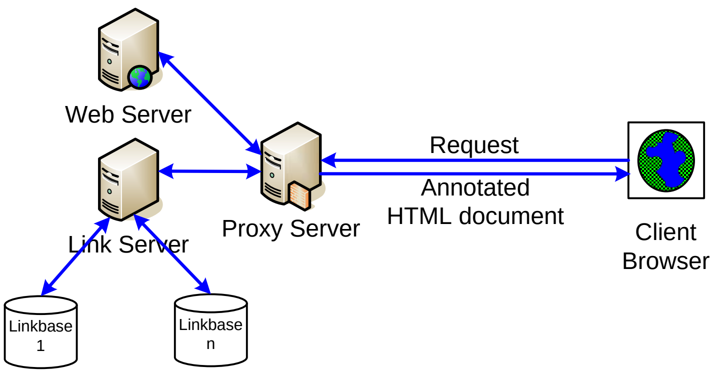


Figure 5-1 The interaction between system environments and the proxy server

To augment a document with links to the content of a web page, a client browser sends a requested URL through the SWLP proxy server. The SWLP proxy server sends a query to the Auld Linky link service (a link server). Then, the link server will return a set of value-pairs of keywords and destination URLs back to the SWLP proxy server. After that, the SWLP proxy server parses the requested HTML file, and matches the keywords received from the link server with the content of the requested file. If the word is matched then the SWLP proxy augments the matched keyword with the destination URL to the HTML file. After all keywords are checked, the proxy server sends the HTML file, which is augmented with the links, back to the client.

5.2.2 SWLP System Implementation

The system provides dynamic links augmentation on web page content; in that a user needed to specify his or her level of knowledge in order to get the system to provide links to learning materials of appropriate difficulty to match the student's level of knowledge.

There are three types of user:

- An administrator is responsible for manage users and courses, and add instructors.
- Learners could register as a new user, manage course registration, and browse and query learning materials based on the level of difficulty required.
- Instructors were able to publish and manage course materials.

Semantic Web used in SWLP

The metadata standards for learning materials have been defined by several communities as given at Learnactivity (ELS). Each community has each own vocabulary. According to guidelines in Ontology Development 101 (Noy and McGuinness, 2001), the reuse of existing ontologies should be considered. The ontology in SWLP is illustrated in Figure 5-2. The metadata standards defined in SWLP are based on widely used existing ontologies which are Dublin Core (DublinCore) to describe any kind of digital resources, and IEEE Learning Object Metadata (IEEELOM) to describe learning materials. Apart from that, the author defines ontology of people for keeping users' information. The namespace of ontology from Dublin Core is *dc*, while IEEELOM is *lom_*. In order for the system to deliver the materials appropriate to a level of difficulty the material is defined by the attribute set in the RDF property for content and context. The content property, defined as dc:subject (like keyword), is what the learning material is about. The context property of learning materials is the presentation of materials (lom_edu:type), such as exercise, figure or example and the difficulty level (lom_edu:difficulty) of materials such as easy or difficult. The ontology is implemented using RDF. The process of querying and navigation through the learning materials is based on this ontology. The ontology is expressive enough to demonstrate how the Semantic Web can be applied in the system.

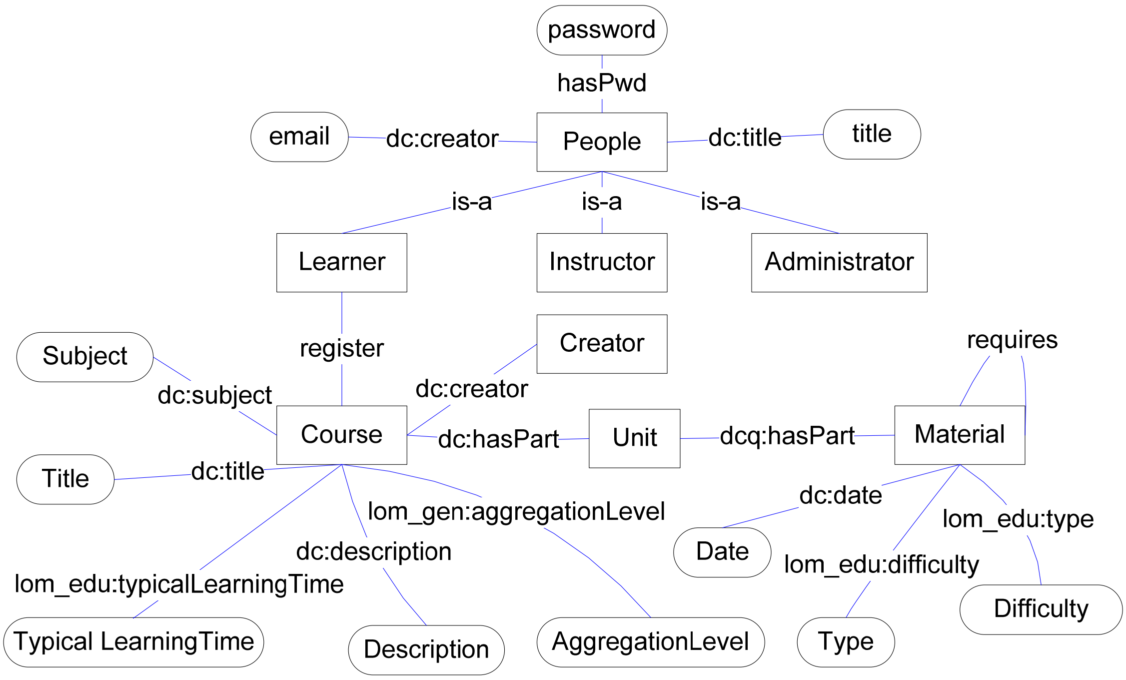


Figure 5-2 Ontology diagram of e-learning portal

SWLP is a dynamic web-based application for controlling all the components of the system which worked together to perform tasks. The system architecture is based on Model, View, Controller architecture[12] in order to support the reusability of model components. The system separates the functions of presentation layer, data layer and business logic layer. Hypertext Markup Language (HTML), Cascading Style Sheet (CSS) and Java Server Page (JSP) are used for presentation. To access the data in the repository and perform the business logic, JavaBeans components are created to perform the tasks.

The controller is used in the business logic layer. Business logic describes the functional algorithms which handle information exchange between a database and a user interface. The controller is responsible for receiving all the requests of the

[12] http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html

applications. For every request that it receives, the controller would decide whether to process the data or display the data. If data needed to be processed such as retrieve data from repository, it would decide to delegate the task to the JSP page containing the required logic or perform itself by calling the suitable JavaBean classes, or it would forward the request to the JSP page which would process the presentation logic.

Accessing the learning materials for an individual user, a semantic query of resources is based on the ontology. Moreover, conceptual navigation through the collection of learning materials is also based on ontological relations. According to SWLP, it would respond to questions such as “I am a beginner to ‘Java’ concept, what learning materials suitable for me?” Figure 5-3 shows a screen shot of the material returned by the system that matches a particular user’s knowledge level . Furthermore, the screen also shows that the document is annotated by the link service (described earlier) with the links, ‘Java’ and ‘UML’, when a client set a proxy of a browser through the Auld Linky proxy server.

In SWLP a simple matching algorithm was used for selecting appropriate learning materials that have difficulty level matched the users’ requests, so a reasoning mechanism or inference engine is not implemented as it is not the requirement of this research. Another limitation was that for the user with a different level of knowledge, there was the problem of what terms or keywords to use when searching for learning materials. For example, the keyword “agent”, could be “actor”, or “performer”. Therefore some mechanism was required to establish shared understanding the keyword. One way to solve this problem could be to use the ontological linking approach as implemented in COHSE (presented in Section 4.5.2).

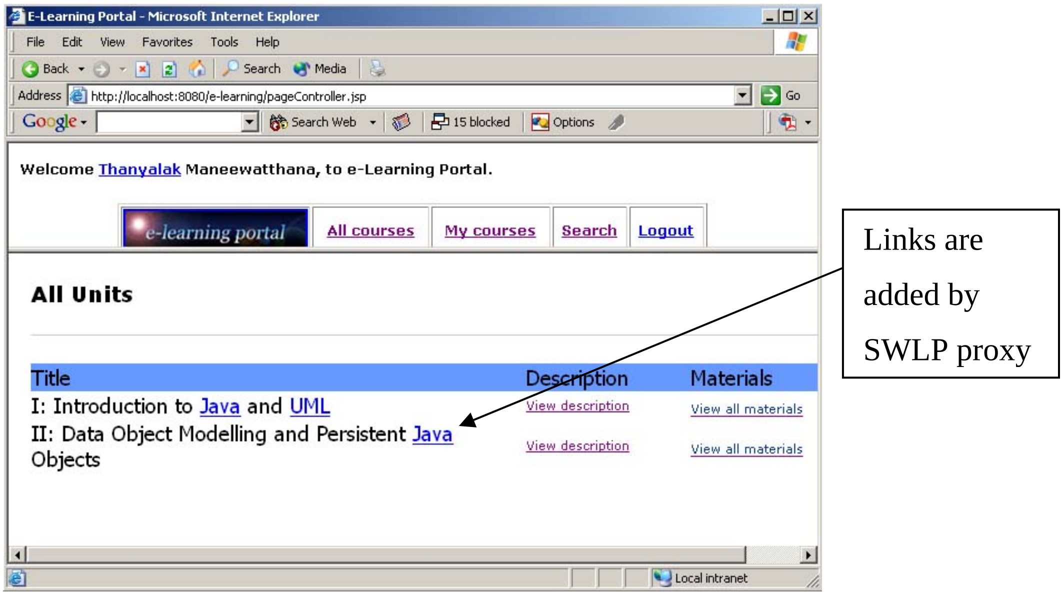


Figure 5-3 A screen shows learning materials when clients set proxy through SWLP proxy server.

5.3 Adaptive Personal Information Environment (*adPIE*)

The aims of the second experiment, were to investigate how the link service approach and the semantic web approach can be applied to support the shareability and reusability of information. The system developed to do this was called *adPIE* .

The Adaptive Personal Information Environment (*adPIE*) aimed to provide a system in which members of the community were able to browse information suitable to their particular requirements, identify and store information in their own information repository, which users may enhance prior to reuse; thereby, enabling the sharing and reusing of structures and data. In addition, *adPIE* enhanced these facilities specify the associations of data and facilitate the interoperability between data components and system. Data and the structure of the information in the system were modelled using FOHM structures (presented in Section 3.5). This experiment has been published in (Maneewatthana, *et al.*, 2005a; Maneewatthana, *et al.*, 2005b).

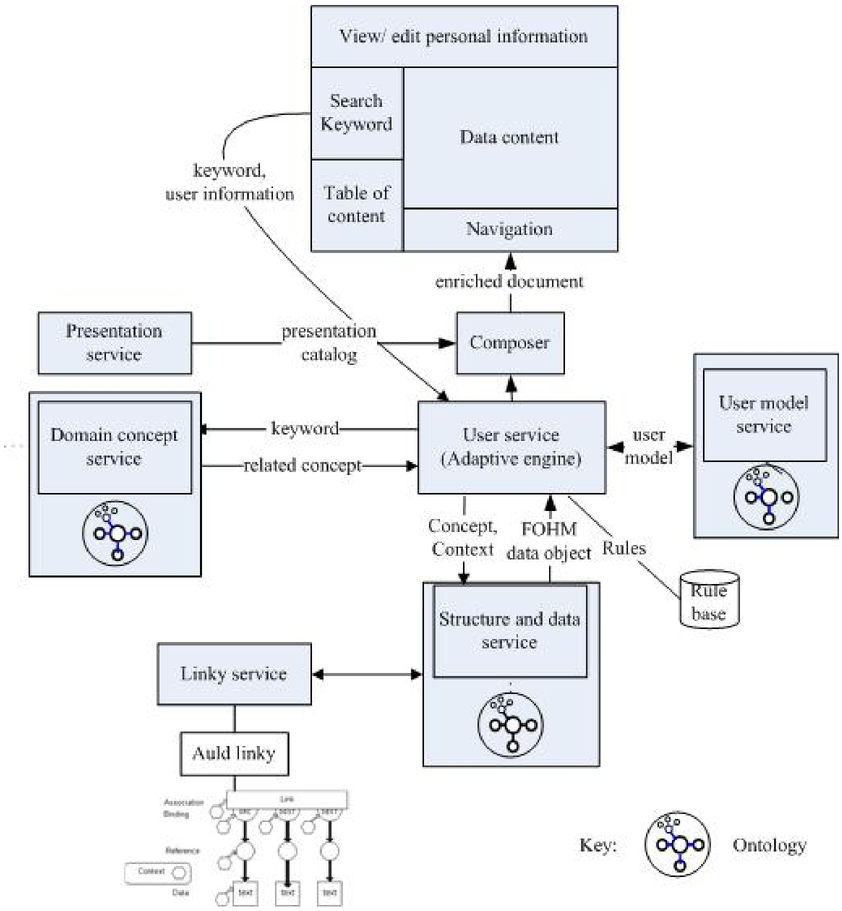


Figure 5-4 *adPIE* System Architecture

adPIE consists of a number of service-oriented components. The domain concept service provides the related concept. The user model service updates user model. The structure and data service manipulates data and structures (associations) as FOHM objects from linkbases through the contextual link server (Auld Linky). The user service or adaptive engine provides the facilities for reconciling the data content, FOHM structures, and user model, to present the individualised document to the user through a web browser.

Figure 5-4 illustrates the system architecture of *adPIE* . The functionality of the system is made available to software agents through a Web Service interface (WSDL), and to end-users by means of a Web browser. The input into the system is a collection of data objects from a user, and the output is an enriched document customised to a particular user's needs.

The following section presents how *adPIE* supports data creation, storage, personalisation, reusability and sharing of information.

5.3.1 Data creation, storage and personalisation

In order to promote reusability and sharing of information, *adPIE* separates the main system components; domain concept, structure and data, user model, and presentation.

- *Domain concept service*: manipulates domain concept used on organisation web sites. Simple Knowledge Organisation System (SKOS) is used to express the basic structure of concept.

- *Structure and data service*: manipulates structure and data represented on the web. Data are represented in the form of FOHM Data objects, while structures connects FOHM data objects defined in XML format as a series of FOHM Association structures. Contexts, represented by FOHM Context objects, can be attached to the Data object, or Association object, for describing the context in which the data item, or association, is visible (or hidden) from the user.

- *User model service:* manipulates user profiles, such as background knowledge, preference, information about user, set of topics, or concepts, where the user has expressed an interest.

- *Presentation service*: provides the display and machine-related information, such as the colour schemes for resource presentation.

- *User service*: is an intermediary engine to integrate every service to work in collaboration.

Structures that can be represented by FOHM is the navigational link which is an association with typed data items, source, destination or bi-directional locations.

*ad*PIE provides adaptive hypermedia support through the use of Auld Linky [Michaelides, *et al.* 2001] as a contextual link server to integrate FOHM structure and data objects according to the context. The context stores value-pairs of concept and level of difficulty of the particular FOHM objects. Therefore, the system can produce the information suitable to users' needs as a personalised web-based system. Once data is made available by the organisation and published on the web site, other users can then use, reuse, or browse, based on user preferences specified in user profiles.

In *adPIE* data are stored in RDF-base. The ontology of the *adPIE* in RDF is shown in Figure 5-5. The ontologies represent relationships of domain concepts. The ontologies are also used to enrich links and data content, and to enable other users to share and reuse the content, or structure through the URI. The user can add data and add the context to particular data objects existing on the Web. A user can add the context, and define whether the context can be seen, shared, or reused (i.e. edited, or deleted) by other users. The user who has permission can then view data with context, as specified by the original user. The users might also use their own domain concept and context for categorising, or describing, the information.

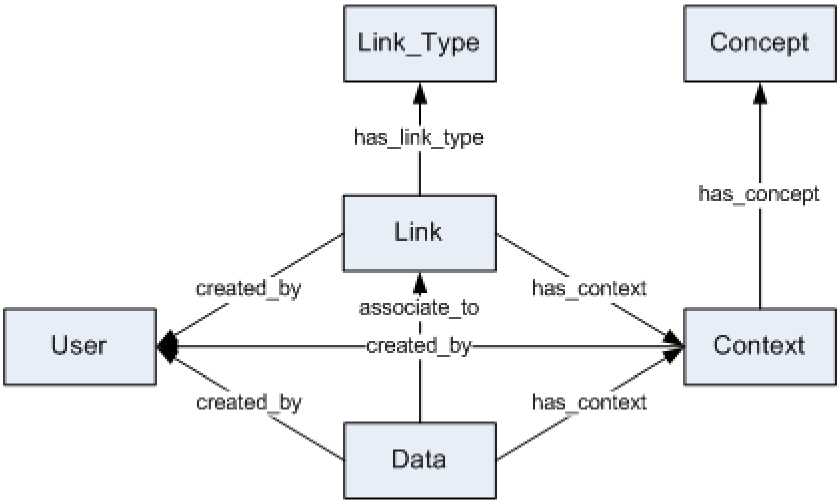


Figure 5-5 The ontology of *adPIE*

Figure 5-6 illustrates a simple scenario on how information can be used and reused in *adPIE*. The browser is divided, apart from menu and navigation areas, into three main areas; search, table of contents and data content areas. The data content area displays the data content stored as FOHM-data objects. In this scenario, a user initiates a query, the results are then returned (after the query processing) to the user according to their profile (the level of detail and types of user). The users add notes, or comments, to a particular piece of information.

The processes for browsing, adding structures the user is interested in, and adding more data content (such as notes or comments) to a particular item, are described in the following sequences of operations (see Figure 5-6).

1. A user enters a keyword into a search box. Then, the system will find the associations that have relationships related to this concept from the metadata of

FOHM associations stored in a RDF file, and then will show the results as table of contents.

2. When the user selects the association they are interested in, the system will get more structure of contents from the linkbase, represented as FOHM associations.

3. The data content stored in FOHM data object will be displayed in the data content area if the user clicks on any item in table of contents. In addition, the user is able to store any information to their personal information area, simply by selecting a particular structure of information, and the appropriate option to include it in their personal information area.

4. The system will manage the storing of the selected association to the personal linkbase.

5. The user is then able to add more data such as notes, or comments, to the data content, which exists on the Web, in the personal information space, while the original data content remains unchanged.

6. The system will save all modification to the linkbases, and the data content that is represented in FOHM data object, in the personal information space.

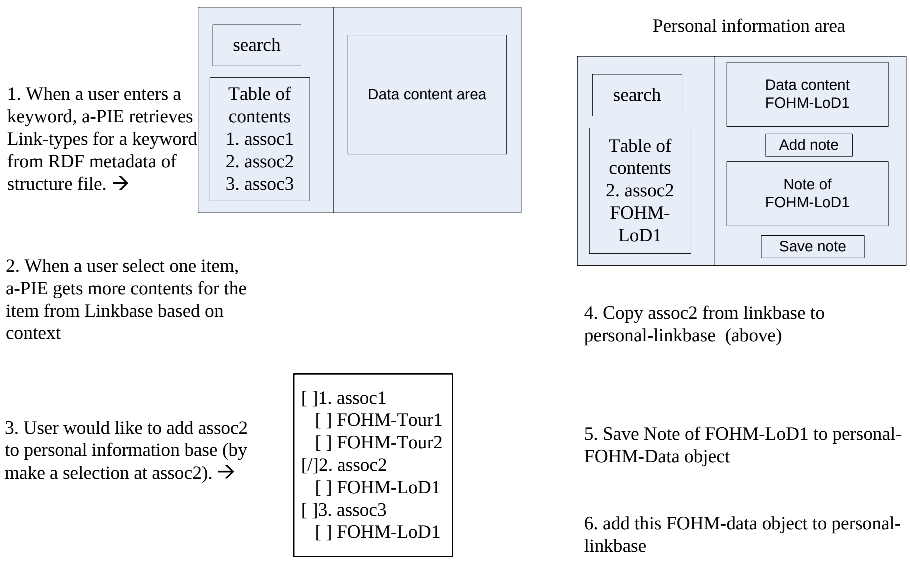


Figure 5-6 Simple processes for adding note to particular data item

5.3.2 Implementation

adPIE is prototyped in Java as a web application under Apache Tomcat, using Auld Linky version 0.72 and Apache Axis. Apache Axis is a java platform for creating and deploying Web Services applications. Java offers various advantages in comparison to other languages, in that it can be used under different operating systems. Another important consideration is the availability of one of the most advanced frameworks to build Semantic Web applications, including a rule-based inference engine which is Jena. *adPIE* uses Jena, an open-source project, to manipulate RDF models, and for a set of limited reasoning features. First prototype was adding note to a particular item and stored in a data object as a link association type.

5.4 Discussion

From the first experiment, the Semantic Web Learning Portal (SWLP), it is shown that from the link service approach, presented in Section 3.6, links can be stored separately and manipulated through a link service. Links can be added as first-class objects to a web page through the use of the SWLP proxy server. According to Bailey, behaviour and context could be added into FOHM linkbases which would have provided some of the adaptation techniques defined by Brusilovsky (Bailey, *et al.*, 2003).

However, De Roure notes (about the DLS proxy) that using a proxy server might not be a good means to annotate links, because users have to manually configure their browsers to ensure that requests for WWW pages are sent through the proxy. In addition, if the user is already using a proxy for some other services it is difficult to chain the proxies together (De Roure, *et al.*, 1999). For example, at a client web-browser only one proxy can be specified.

According to the second experiment, ad-PIE, there are two issues found in the *adPIE* prototype. One is a bootstrap issue because of using Auld Linky for parsing FOHM model in the a-PIE. With Auld Linky, the linkbase has to be loaded into memory prior to serving links, therefore when there is a new data object added to FOHM linkbase the Auld Linky has to be restarted. In addition, a context object is not flexible, in that in a case where there are several context objects with similar keywords but different values (as key-value pair), Auld Linky delivers only the data object that matches the last context object. For example, when there is the same key in context objects attached to two pieces of different learning materials; Material 1 and Material 2 see Table 5-1.

Data Object	**Context Object**
Material 1	(keyword = 'Java') (keyword = 'Agent')
Material 2	(keyword = 'Agent') (keyword = 'Image')

Table 5-1 Two context objects with the keyword on different material

If the system would like to get learning materials with the keywords 'Agent', both materials are supposed to be displayed. However, when Auld Linky queries the FOHM structure with a context; (keyword = 'Agent') only Data object: Material1 is returned.

5.5 Summary

This chapter has described early prototype systems to investigate the limitations of the Semantic Web technologies in supporting the sharing of content and links in adaptive hypermedia systems. A link service, ontology manipulation and the Semantic Web-based application have been investigated. In SWLP, the proxy sever was implemented and invoked the Auld Linky link service. The experiment has demonstrated that it is possible to integrate the link service, adaptive hypermedia and the Semantic Web technology together to provide the relevant information to particular users' needs. However, the proxy approach might not be a right way to manipulate annotations to support research motivation in Section 1.2.

The purpose of the Second experiment, *adPIE* , was to demonstrate how the Semantic Web, the link service approach and adaptive hypermedia can support adaptable annotation and personal information environment. Two issues have been found in this prototype; which are the flexibility of context object described in FOHM and bootstrap limitation of using Auld Linky as a link server. Due to these limitations a new framework for contextual annotations is required along with a new method of implementation; this is discussed in the next chapter.

Chapter 6 Contextual Annotation Framework

6.1 Introduction

The previous chapters presented the relevant background and experiments for this research. This chapter presents the Contextual Annotation Framework (CAF) with the contextual annotation concept, which utilises the semantic web technology, open hypermedia and adaptive hypermedia principles to provide contextual annotations. It describes how the background has impacted on the design of the CAF approach to a web-based contextual annotation system, the contextual annotation concept including the issues concerning the design of the framework and storage models.

The motivation of this thesis is given in Section 1.2, while requirements for developing a new framework will be provided in this chapter. First, the chapter presents the definitions that will be used in this chapter and then discusses the related systems. It is followed by the requirements of the framework. Later, the chapter concentrates on the proposed framework which provides an environment for a web-based contextual annotation. Finally it concludes with the mechanisms and implementations of components in the framework, how annotation works in order to provide adaptable annotations, and how the framework supports sharing and reuse of annotations.

6.2 Definitions

In this research, the definitions for terms used are the followings.

Annotate: is to make or furnish critical or explanatory notes or comment, or to make or furnish annotations for (as a literary work or subject) (Webster, 2005).

*Annotation***:** a note added by way of comment or explanation or the act of annotating (Webster, 2005).

Annotation object: is any object such as notes, comments or ideas that are attached to a web document or selected part of the web document (Koivunen, 2001).

Annotation type: is the relation of annotation object related to the annotated content such as comments, examples and explanation. It represents the relationship between annotation object and the annotated text.

Context object: is a complex object containing information related to the context of the annotation object. The context object is used to define conditions for the visibility of annotation objects, for example, context of user's work (location, time, and platform), expertise level to understand the annotation object (beginner, intermediate and advance), or other factors that are needed to properly interpret an annotation object.

Contextual annotation object: is a complex data structure that comprises one annotation object plus one or more context objects.

6.3 Revisiting the Related Work

While, Section 4.4 discussed how people gain knowledge and Section 4.5 presented the ideas that writing annotation objects while reading was a vehicle for critical thinking and learning, and organising information and knowledge. Section 1.2 showed research motivation. Two issues that this research addresses are shareability and reusability of annotation objects, and information overload in terms of annotation objects.

Different people have different preferences or interests and each annotation object might only be related to some contexts such as location, concept and expertise level. Figure 6-1 illustrates annotations without contexts. The upper diagram shows where there is no context for annotation objects; consequently all annotation objects will be delivered to users causing information (annotation) overload. In Figure 6-1 the lower diagram shows where there is an annotation type, such as explanation, example or definition, of annotation object for an annotated area; however, all annotation objects for a particular annotation type will be presented to users no matter what context, such as topic or expertise level, of that annotation object is about.

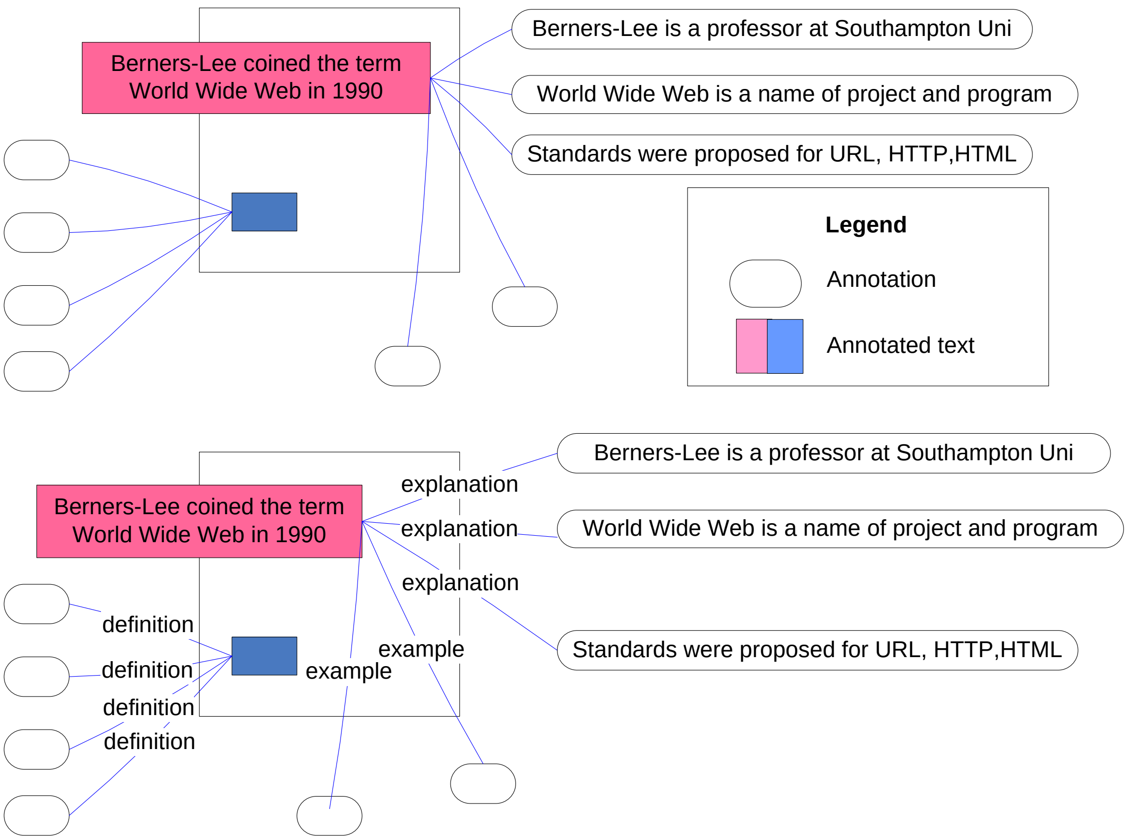


Figure 6-1 Annotations without context

Researchers in adaptive hypermedia (AH), described in Chapter 2, are concerned about the information overload problem. The research in AH has been focused on the formal models and systems, and frameworks for AH systems. The formal models (presented in Section 2.7.2), such as Dexter, AHAM and Munich, are used to express the functionality of an adaptive hypermedia system but have no mechanisms for implementing it. Adaptive hypermedia systems and frameworks aim to provide adaptive hypermedia in a particular domain, for example e-learning and e-commerce, however, not for annotation systems.

Useful annotations can be invaluable knowledge for other people in a community. Therefore it is beneficial if annotations can be shared with others. Generally hyperlinks in web pages are hard-coded and embedded in the content of the document. Therefore links are difficult to author for anyone but the original author. For example, uni-directional and binary links can be created and these links must be embedded in the web content created by their authors (Anderson, 1997). However, research has shown that open hypermedia technology (presented in Chapter3) can be used to enrich the web hypertext model. XML Linking (XLink) (DeRose, et al., 2001) allows elements to be inserted into XML documents in order to describe links between resources.

6.4 The Requirements for a Framework

To summarise the main points, to provide a system environment for the scenario as a research motivation given in Section 1.2, the main properties of a contextual annotation system should:

- deliver contextual annotation objects based on users' preferences
- allow users to specify the annotation type; the relationship between annotation objects to the annotated area.
- enable users to store other people's annotation objects in their own repository.
- allow users to specify context to an annotation object while the context value is any condition for the visibility of the annotation object to an individual user.
- allow users to change the context of others' annotation objects and store them in their own repository.

In order to support all of the properties presented above and the scenario given in Section 1.2, a web-based annotation system should have the following characteristics.

Adaptable Annotation: An annotation system should be an adaptable system that supports personalisation by delivering annotation objects suitable for user's needs, in order to provide information relevant to the user's profile which should help users to find information much easier and faster.

Annotation with context: in order to ensure that the annotation systems deliver relevant annotation objects to users, the annotation object should provide a method that allows users to add context values to each annotation object. This is achieved by using a context object as part of the annotation.

Annotation manipulation: annotation objects should be separated from web documents in order to be transferable and independently manageable (added, edited, deleted).

Semantic interoperability: annotation objects are reusable when the meaning of annotations is expressive enough to be shared across systems. A framework has to ensure that the intended semantics of annotation objects can be shared among different systems.

Syntactic interoperability: in order to share annotation objects across framework, the annotations should have a formal syntax. The semantic web is an important means of annotations. A semantic description of the annotation objects should be defined in a web language, such as OWL, RDF(S)/XML.

According to Table 4-2, most annotation tools do not support all of these properties. These systems or frameworks have no context objects to the annotation objects, therefore all annotations will be presented to users. For instance, Annotea and AWF are annotation frameworks that give all the annotations for a particular URL. Although ComMentor (see Section 4.5.1) provided typed annotation objects, it did not allow users to organise annotations, did not allow users to store other's annotation objects, and did not research utilising the Semantic Web technology or the ontology. Based on these requirements, a new framework is proposed to support these requirements.

6.5 Proposed Framework: Contextual Annotation Framework

The objective of this thesis is to propose and evaluate a contextual annotation framework (CAF) approach for implementing a web-based contextual annotation system. The goal is to ensure that the proposed approach facilitates a system which serves all requirements in Section 6.4.

6.5.1 The components of the framework

In response to the information overload problem in terms of annotation objects in a web document, the annotation system should not provide too many irrelevant annotation objects to readers. Instead, as mentioned in Section 2.4, users should get only relevant annotations based on their preferences.

The contextual annotation framework proposed is a personalised system. Therefore in addition to the domain model there is a user model, an adaptation model and an adaptation engine. These are integrated as components in an adaptive hypermedia system, such systems were presented in Section 2.6. In the contextual annotation framework annotations are filtered by using cognitive and social approaches (described in Section 2.4). Cognitive filtering in the proposed framework means the annotations will be filtered based on users' requirements, while the annotations will be ordered by the frequency of popularity of annotations being added to others' repositories.

The user model in the CAF is persistent in that the user information or profile will be kept over time. The annotations are manipulated by an individual user; at the same time they can be shared with users in similar groups. User information is extracted explicitly from user behaviour by enabling users to set these themselves. User model perspective as a component in adaptive hypermedia system is given in Section 2.6.

6.5.2 The contextual annotation object

In order to provide the right annotation objects to the right person, the *contextual annotation object* is proposed.

A context object is as an important object for determining the visibility of an annotation object in the annotation-base. Context, such as users' preferences for a topic and their expertise level in a particular topic, can be used as criteria or conditions to cull irrelevant annotations. A context object can be used in the personalisation mechanism. The context object is attached to the annotation object and indicates that users only want this annotation object to be viewable if the context, specified in users' profiles, matches the context value in contextual annotation objects. In other words, the context object is a modelling decision. One annotation object can have several context objects because one annotation object can be used in different contexts. In a situation where there is more than one condition in a context, not only the topic but the level of background knowledge for that particular topic is also important in order to understand that topic. In this approach each condition is considered as one condition or criteria and all criteria are formed for a context object, as a complex object.

The association structure of contextual annotation objects should be organised into meaningful groups using annotation type. For example, contextual annotation objects that present the relationship 'explanation' of an annotated region can be collectively put under the same 'association' object. Therefore, in one association there can have more than one contextual annotation object.

Figure 6-2 shows that only annotations with relevant contexts will be presented to users, while Figure 6-1 showed all annotations being given to users.

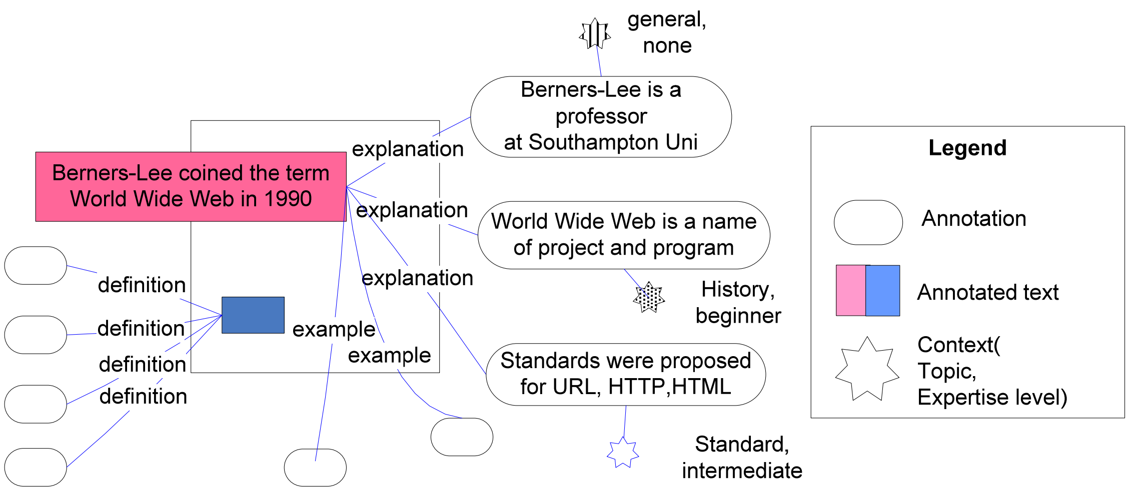


Figure 6-2 Annotations with contexts

6.5.3 *Shareability and Reusability of Contextual Annotation Objects using OH and Semantic Web Approach*

To achieve shareability and reusability of annotations, the CAF system should provide support for semantically described hyperstructures (and a method for an author to handle annotations) that are stored separately from documents. The annotation should be expressive enough to capture the meaning, structure, description and context in which the annotation exists. This is achieved by adding metadata to annotations to enrich them. As with links in open hypermedia, in this research the annotations are structured and manipulated as first-class objects (Wills, *et al.*, 2003) by storing annotations separately from web document contents, in an annotation-base, and these annotations are managed through the ontology-based contextual annotation service.

Ontologies represented in the form of RDF(S)/OWL have been used as a backbone of the CAF to semantically define explicit relationships between entities, annotation metadata, user and personal repository, and domain concepts. In order to facilitate applicability and wide deployment, the system should use open and standard web infrastructures (Vasudevan and Palmer, 1999). This framework is based on these requirements, therefore it is possible to share and reuse annotations not only within web applications, but also across web applications.

Figure 6-3 illustrates the CAF approach where contextual annotation objects will be categorised according to the annotation type, such as explanation, from Figure 6-2. The location of source URL, and annotated text, is stored ("src"). The contextual annotation objects are destination ("dst"). These contextual annotation objects explain the annotated region. There can be several contextual annotation objects in one annotation type (shown in the figure as "explanation"). Each annotation objects has one or more context objects. In one context object there can be several conditions or dimensions to provide more relevant annotation objects to users. Because these models are based on RDFS/OWL, it is possible to reuse these models and share every part in contextual annotation objects; annotation objects, associations or context objects, by using URIs.

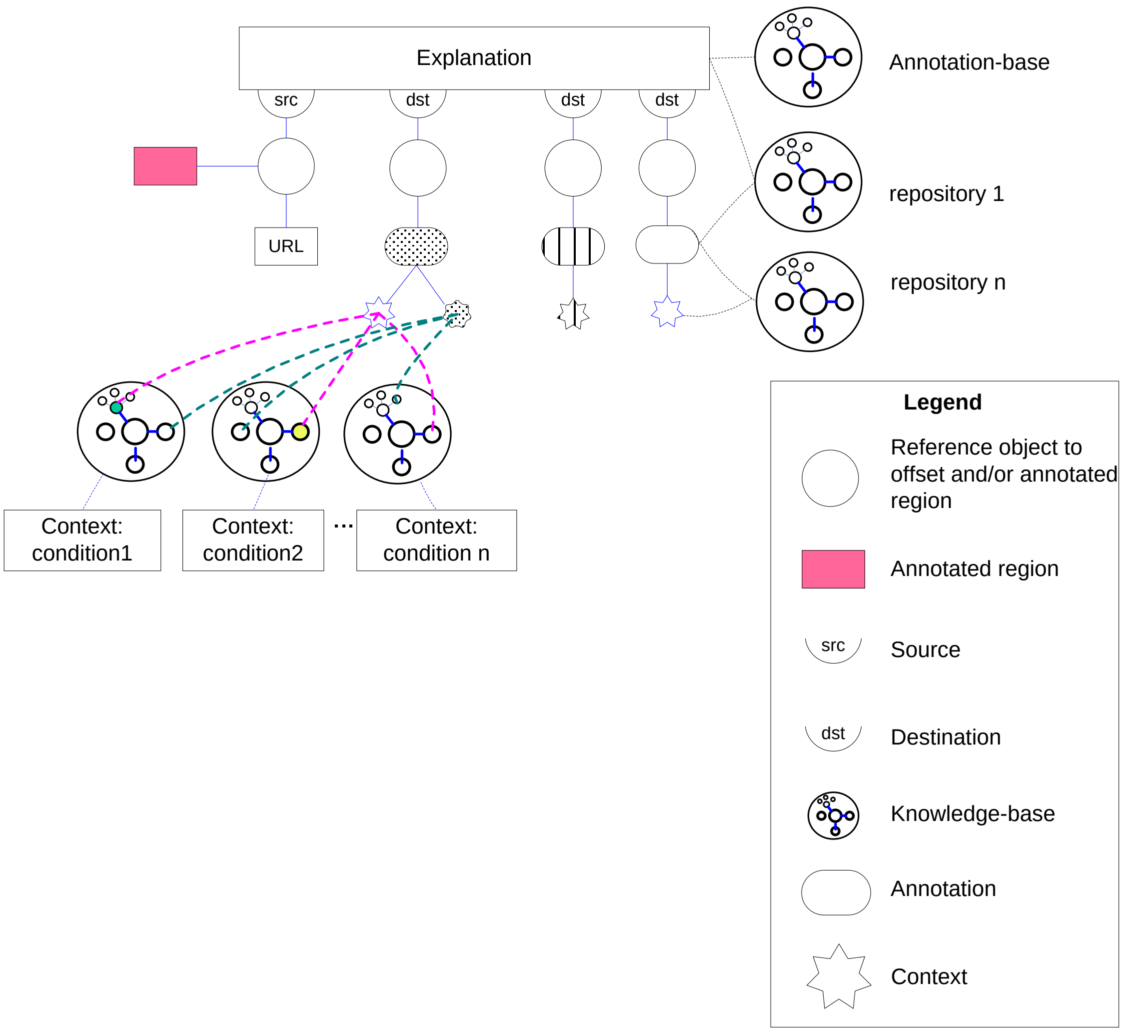


Figure 6-3 Sharing and Reuse of Objects in the CAF Approach

In order to assure reusability, the framework has taken the viewpoints of Lowe and Hall (Lowe and Hall, 1999), and Robson (Robson, 2004) by separating structure, content, context and presentation.

6.5.4 The Novel of the Thesis

To summarise, the novel nature of this thesis, the Contextual Annotation Framework (CAF), using the Ontology-Based Contextual Annotation Service, is presented. The contextual annotation concept, the annotation objects with context objects, is introduced. The innovation is the framework to provide annotation system as an adaptable web-based system allowing users to personalised annotation objects based on users' profiles. It also permits users to annotate information, share and reuse annotations by enable contextual annotation objects transferable, for example, allowing users to keep contextual annotation objects created by others in their own repository and set personal context to them. The ontology, in RDFS/OWL, of human-created annotation used in the framework has been designed more formally and implemented them in an open hypermedia framework. The ontology-based contextual annotation service is implemented as an annotation service (functions like a link service) for managing and delivering contextual annotation objects which are stored separately in the annotation-base. The research proves that it is beneficial to use the contextual annotation framework based on Semantic Web technology with the ontology-based contextual annotation service approach to provide adaptation of annotations, and sharing and reuse of annotations.

6.6 Developing the Framework

The CAF framework consists of four core layers separately in order to support shareability and reusability. These four layers are application layer, services layer, metadata layer and resources layer.

- *Application layer* is for a particular domain or application.
- *Services layer* provides services for an application to manipulate resources.
- *Metadata layer* stores metadata for information, including domain resources specific to particular domains that are used to structure the data and categorise resources, and all kinds of associations. These resources are represented as ontologies using RDF(S)/OWL.

- *Resources layer* stores all resources represented in RDF using URI as identifiers.

Developing the CAF involved identifying the major components, which are the system components and the storage models. System components, in the application layer and services layer, provide interfaces to handle requests and responses, and to manipulate resources in the resources layer through the ontology in the metadata layer. Figure 6-4 illustrates the layers of contextual annotation framework.

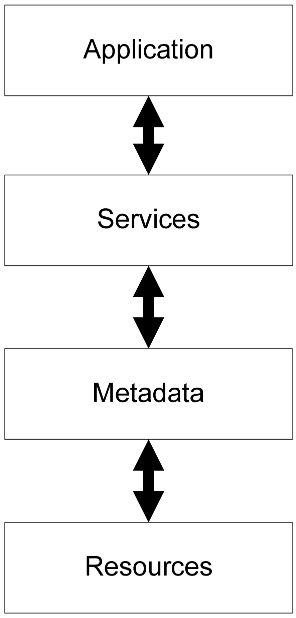


Figure 6-4 Layers of Contextual Annotation Framework

6.6.1 System Components

The system components consist of: an intermediary, annotation services, a user service, a domain concept service, an ontology-based contextual annotation service, and a context service. Only the intermediary is in the application layer, presented in the previous section. All other services are in the services layer. Figure 6-5 illustrates the conceptual view of these system components.

The Intermediary

The intermediary, in the application layer, acts as the glue between the user (client-side) and the service-side services. The intermediary has two roles. First, it is responsible for every aspect of the client requests. This includes obtaining user information and user annotation requests. Secondly, the intermediary responds to user requests, sending profile data to the annotation services, formulating queries to services on the service-side and presenting the results to the user.

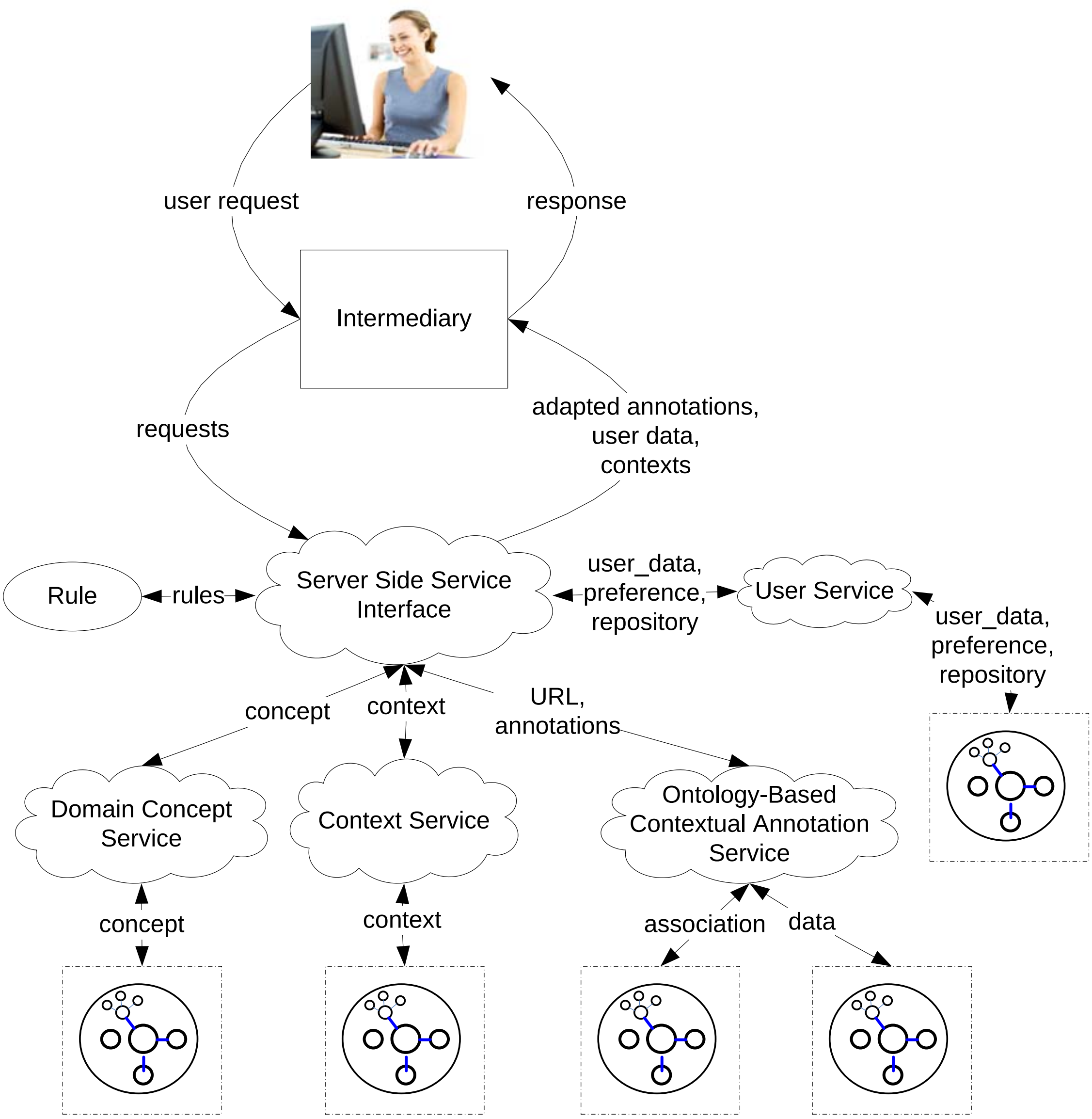


Figure 6-5 Components in the CAF

The Server-Side Service Interface

The server-side service interface handles requests and responses between the intermediary and related services: user service, domain concept service, context service and ontology-based contextual annotation service. It responds to requests from the user (through the intermediary) and invokes other services to perform tasks. It also applies adaptation rules to these services to produce personalised annotations. These rules may be stored in databases, sometimes referred to as rule-base (Brusilovsky, 1996b), or hard-coded into the application.

The User Service

The role of the user service is to store and process the user profile (user information and interests), and manipulate, create, delete and modify user information and personal annotations in a personal repository. Although security is not examined in this research, the user service component of a public system would have the responsibility for protecting and maintaining user privacy, and providing authentication mechanisms.

The Domain Concept Service

Concepts can be organised in hierarchies using broader-narrower relationships, or linked by associative relationships (Miles and Brickley, 2005). The domain concept service role is to add, modify and delete concepts used in the application domain. In addition, it provides concept definition and related concept attributes, such as narrower concept or broader concept, requested from the user service (for manipulating user's interests), the context service and the ontology-based contextual annotation service (for providing personalised annotations).

The Ontology-Based Contextual Annotation Service

The ontology-based contextual annotation service handles requests and responses for creating, removing and modifying annotations with contexts as links (described in Section 6.2).

The Context Service

The context service role is to provide mechanisms for manipulating, creating, modifying and removing the context of annotations. The contexts served by the context service are used together with the ontology-based contextual annotation service, the domain concept service, and the user service.

6.6.2 Storage Models

The storage models consist of: association structure model, context model, data model, user model and repository model. These models are defined as ontologies represented in RDFS/OWL. The ontology is developed to provide a shared common understanding of these models among people or software agents and enable reuse of knowledge in a domain. The ontology of the framework provides interoperability among the system components presented in Section 6.6.1. The instances of the ontology, considered as knowledge (Noy and McGuinness, 2001), which are used in the system, are stored in an RDF-base. Each instance has a URI for reference, therefore it is easy to integrate, share and reuse in an application. Following the guide written by Noy and McGuinness (2001), the ontology development in this research is presented here.

Developing the ontology

As a first step, the domain and scope requirements have to be identified. In this framework, the concepts describe the association of annotations, domain concept, data, context and user information. Requirements of the framework are providing suitable annotations for user's preferences, and allowing users to keep a particular annotation and change context prior to storing in a personal repository. Competency questions have to be specified, which are used to determine the scope of the ontology that the knowledge base should be able to answer. In this framework, some sample questions for this research are given below.

1. For a particular URI, which annotations should the system produce to a user when the user is a 'beginner' to the 'web developing' concept?

2. Which annotations should the system produce to 'explain' the 'colour-scheme' concept for an 'intermediate' user?

3. What annotations are of interest to the user (they are stored in that user's repository)?

4. Is that user allowed to see this annotation?

5. What annotations are considered useful or interesting (because they are stored in some repositories)?

Reusing Existing Ontologies: SKOS, FOAF, Dublin Core

The next step is to consider reusing existing ontologies. Some of Friend Of A Friend (FOAF) ontology (Brickley and Miller, 2007), Simple Knowledge Organisation System

(SKOS) (Miles and Brickley, 2005), and Dublin Core (DCMI, 2006) are reused. Since the FOAF ontology is widely used to express people and others, the FOAF document and FOAF name are used. SKOS ontology is used to define the domain concept; prefLabel, broader, narrower properties are adopted, as SKOS ontology can express the basic structure and content of concept schemes and widely adopted. Dublin Core ontology is a general-purpose ontology. Therefore subject, title, date and creator attributes from the Dublin Core ontology are used to describe annotation as metadata.

Reusing Existing Ontologies: FOHM

As FOHM is a model that is suited to the requirement of this framework, an ontology of FOHM (Gibbins, *et al.*, 2003) is adopted as a first pass at describing the structure associations. This FOHM ontology has been expanded to fit the requirements of the CAF framework.

FOHM supports hyper structure with contexts and behaviour objects, in other words FOHM has standard navigational and document interpretation, whereas RDF is only a data linking language. Another reason that FOHM's ontology is adopted is that the categorisation process can be easily refined by adding additional context objects. In general, the context of any web application can be any object which can be identified by the URI as a resource of an RDF triple (described in Section 4.3).

Listing important terms, defining classes and classes hierachy

The final step is to list important terms, define the classes and the class hierarchy, including properties of classes and restrictions for each property. The main classes of the framework are: association structure, context, concept, data, user, and repository. Each class is described below. After performing these steps iteratively, the ontology is ready to be applied and create instances in order to build a knowledge base. OWL DL has been designed on top of RDF(S) to define the ontology. RDFS has been used to define subclass, domain and range of properties while OWL DL has been used to declare restriction and cardinality of properties. More detail of ontology is presented in Appendix C.

The Association Structure, Context, Data, Concept, User and Repository Models

The Association Structure, Context, Data, Concept, User, and Repository models are now presented. The ontology has been created using RDFS/OWL. The cardinality is also presented. The prefixed namespace of ontology that are used in the relationships are the following.

> *fohm* and *nav* are ontologies defined by Gibbins (Gibbins, *et al.*, 2003). The former is a namespace of the FOHM ontology, while the latter is a namespace for navigation association ontology which is a subclass of FOHM Association.
>
> *dc* is a namespace of Dublin Core ontology.
>
> *foaf* is a namespace of FOAF ontology.
>
> *skos* is a namespace of simple knowledge organisation system.
>
> *caf* is a namespace of contextual annotation framework ontology that was created for this research.

Association Structure Model

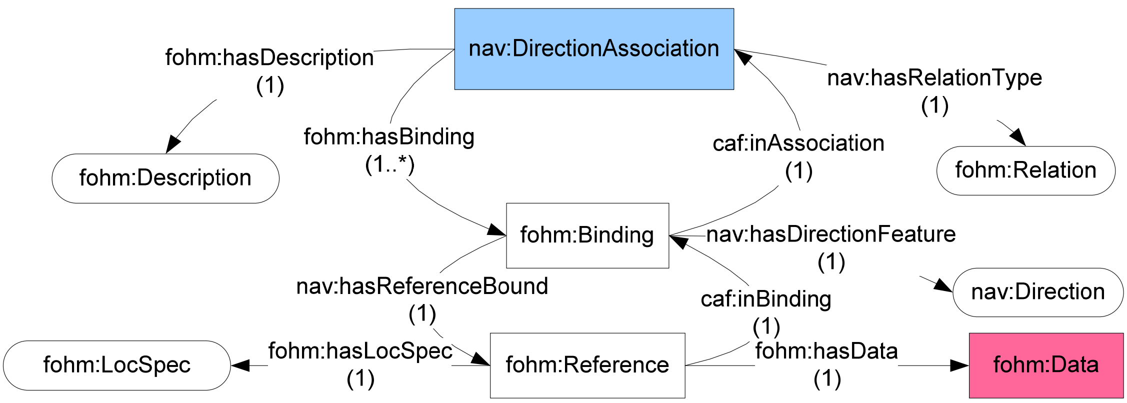


Figure 6-6 Association Structure

The association structure model (defined as the DirectionAssociation class) is a navigation association which is a subclass of FOHM Association objects. The association itself has a description (fohm:hasDescription). It also has a relation type of a particular association between an annotation and the annotated region. The instance of a relation can be defined in a Relation concept (fohm:Relation), for example ‘explanation’ or ‘summary’. The direction association model represents relationships between a source data object and a destination data object. The association object binds (fohm:hasBinding) to a data class (fohm:Data) according to a direction, that is defined by nav:DirectionFeature such as source, destination or bidirection, through a reference class (fohm:Reference) at the location of the annotated region at the source document (fohm:LocSpec). Figure 6-2 shows the annotations with context and relation type (for example: 'explanation', 'definition') which specify the relationship between the annotated region and annotations, while Figure 6-3 shows annotations in the 'explanation' annotation type represented in the FOHM model.

Context Object Model

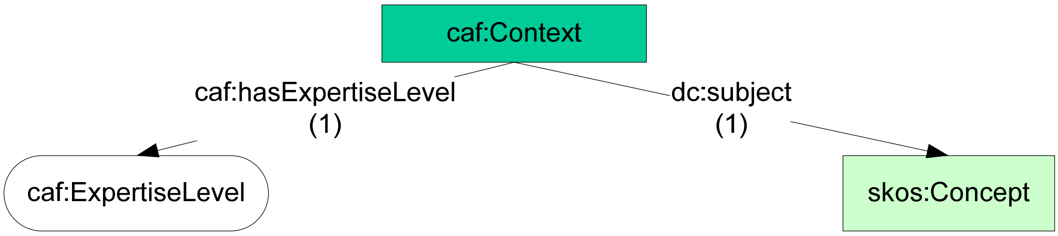


Figure 6-7 Context Model

Context object model (caf:Context class) is an important entity for adaptation. In this research the context consists of two conditions: concept and expertise level. However, language might be used, together with concept and expertise level, to specify the language of particular information. This is an example of a context consisting of: concept, expertise level and language.

The context object is used by the ontology-based contextual service to decide which annotations should be given to the user. The context model can be any object which can be identified by its URI. In the prototype, the context can have only two dimensions that users prefer: a concept (skos:Concept) and an expertise level (caf: ExpertiseLevel).

Data Model

The data model (fohm:Data class) can be a source anchor or a destination anchor of an annotation. The fohm:Data stores content or a URL or both. The data model stores annotated content, while a reference object presented in the association structure model stores the location of annotated content. When a user creates an annotation, the URL of the web being annotated is stored as a source of the link, while the annotation body is stored as a destination of the link. Metadata which are created date (dcterms:created), modified date (dcterms:modified) and user (dc:creator), are generated from the system. The data model has to specify which group of users (caf:authorisedGroup) can see this data object. In addition, in order to support adaptability, the data model has to specify at least one context (caf:hasContext).

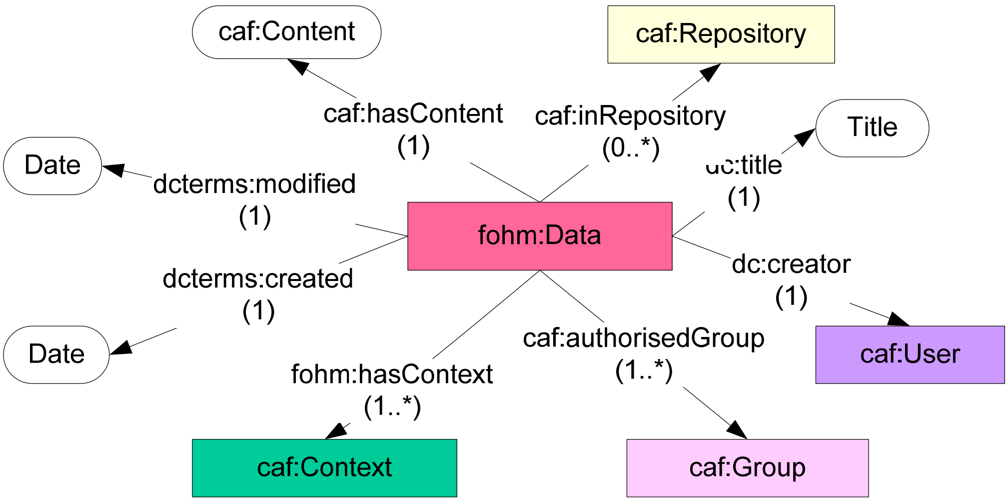


Figure 6-8 Data Model

User Model

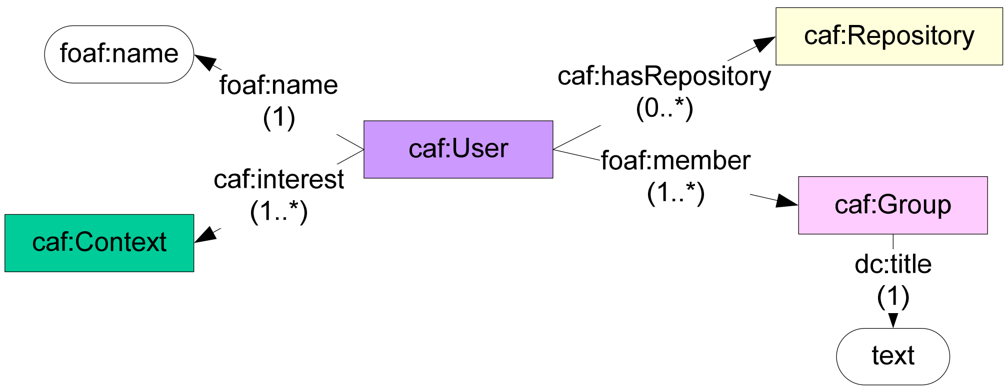


Figure 6-9 User Model

In CAF, the user model (caf:User) stores personal profiles (in this framework, only the name (foaf:name) of the user is stored) and individual preferences (caf:Context), but in public systems there could be other attributes such as email address and full name. Every user has to be a member of at least one group (such as public group, or specific group name foaf:member). In addition, a user can own interesting annotations in a personal repository (caf:Repository).

Repository Model

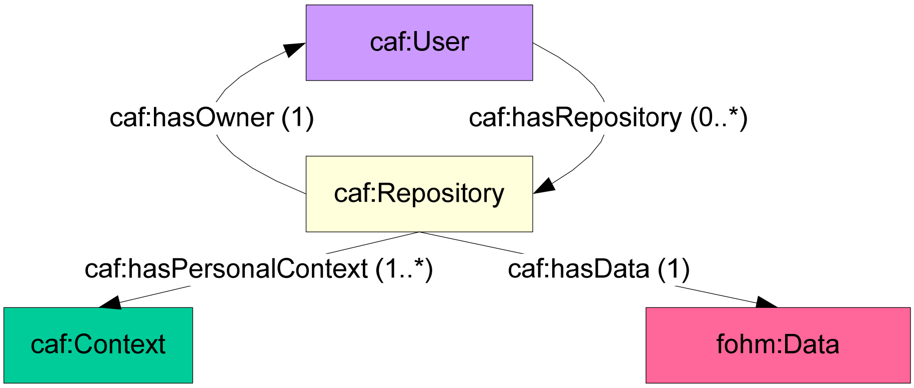


Figure 6-10 Repository Model

The repository model (caf:Repository class) owned by a user (caf:User) stores a data object (fohm: Data) which is an annotation, and personal contexts (caf:Context).

In this research these models form an arrangement as shown in Figure 6-11.

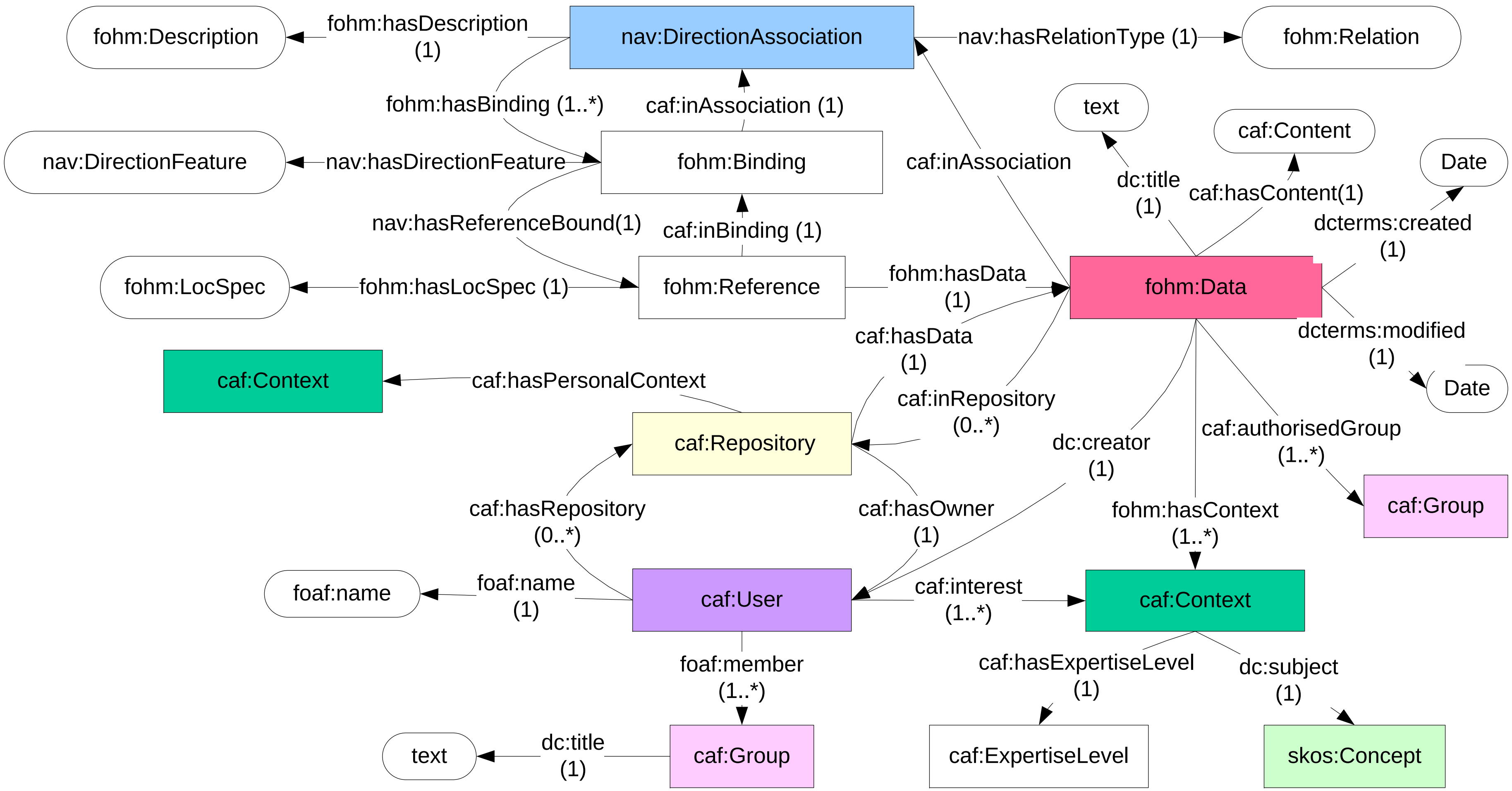


Figure 6-11 Ontology of the Framework

6.7 Technologies supporting the Prototype

Web annotation is a useful mechanism that can support web document management applications. Web annotation systems that are lightweight, platform-independent and scaleable are needed since the number of web pages grows rapidly (Vasudevan and Palmer, 1999). The CAF has been implemented as a prototype called an **annotation and personal information environment (a-PIE)** to support the scenario. The implementation of the components of the framework is based on a service-oriented approach for the following reasons. First, the service-oriented and modular architecture should be the future direction of adaptive hypermedia systems in order to provide shareability and reusability of components (Brusilovsky, 2001; De Bra, *et al.*, 2004). Secondly, having a loosely-coupled system provides more flexibility for introducing new functionality into the system such as implementing adaptive content presentation. The technologies supporting the prototype are presented.

6.7.1 Server-Side Scripting Language

In this implementation, PHP (www.php.net) is used as a server-side scripting language. In a-PIE PHP is used for both client-side and server-side services.

6.7.2 Data Representation using the Semantic Web technology

RDF(S) and OWL, presented in Section 4.3, provide a framework for representing information in a structured and machine-understandable way. The models represented with RDFS/OWL may be exposed outside the system enabling their reuse. For example, the lists of annotations can be read and interpreted by any RDF/OWL parser.

6.7.3 Data Storage

Data can be stored in either a database management system or text file so that the RDF/OWL parser can process it. However, storing knowledge in a database is more convenient than in text files since users do not need to handle the bootstrap function. In a-PIE, MySQL (www.mysql.com) is used to store models.

6.7.4 Data Parsing and Querying

Models in a-PIE are represented in RDF/OWL. Therefore a parser is needed for manipulating these models. RAP[13], a Semantic Web toolkit for PHP developers, is used to handle these functions. RAP provides a statement-centric API for manipulating RDF graphs as a set of statements and a resource-centric API for manipulating RDF graphs as a set of resources. RAP integrates RDF/XML parsers and serialisers. It supports in-memory or database model storage.

6.7.5 Web Services

REST[14] web services means that each unique URL is a representation of some object. The main advantages of REST web services are first that it is lightweight which needs no extra XML mark up, like SOAP or RPC. Secondly, it provides human readable results, and lastly it is easy to build therefore there are no toolkits required.

The a-PIE prototype has four operations for each service: getting a list of models, adding a model, deleting a model, and modifying a model. REST is suitable for these operations. The annotation and other functions on the client-side, coded in JavaScript or scripting language such as PHP, can communicates over XMLHttp to these server-side services.

6.7.6 Server–Client-based Annotation

There are several client-based annotation systems that plug into the browser. Some are commercial, others not. There is great resistance to installing client-based software. Server-based annotation is adopted in the a-PIE prototype.

6.7.7 Personalisation

In the prototype, the static user profile was used. The static user profile implementation involves presenting the user with categories and subcategories of concepts and also expertise level as the second dimension of context of a particular concept, and asking

[13] http://sites.wiwiss.fu-berlin.de/suhl/bizer/rdfapi/
[14] www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

the user to select the topic in which they are interested. Once this is done a profile is built up and stored. The profile is simply an indication of the categories in which the user has shown an interest. Once the preferences are submitted they cannot be changed by any mechanism. The only way to redefine the interest is by going through the interest declaration process and redefining the interest. Therefore, interest definition is an explicit exercise undertaken by the user. The user model can be used to individualise the annotations.

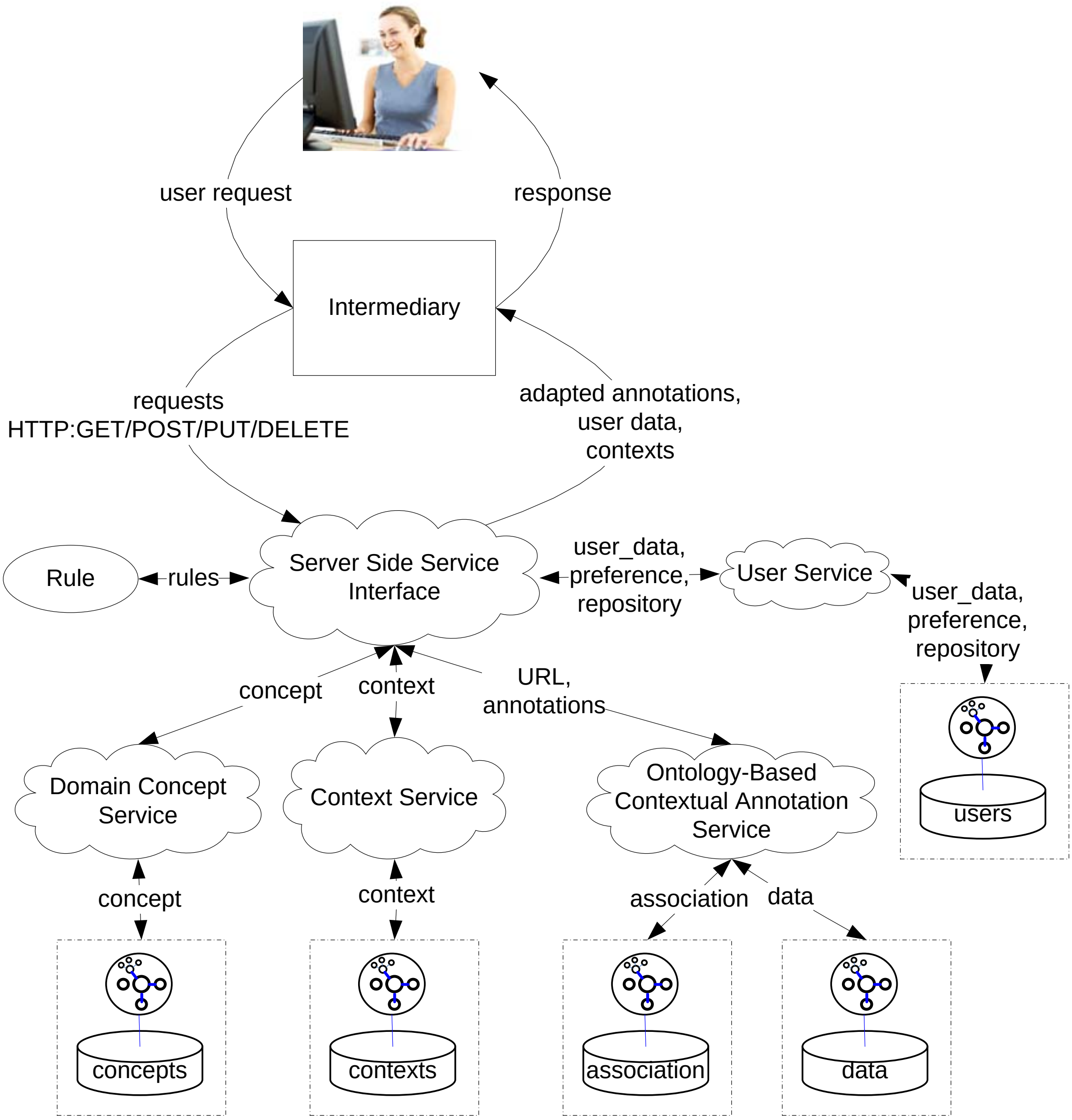


Figure 6-12 System components of the a-PIE

6.7.8 The Ontology-Base Contextual Annotation Service

The annotation service was prototyped as a personalised engine for filtering annotations from the annotation-base according to context objects being matched. The characteristics of the service are the following.

- It supports querying annotations for a URL with context and group.
- Context can be any object with a URI, and there can be many items but in this case the level of expertise and concept are the contexts.
- Architecture: the separation of the models into separate repositories supports their reusability.
- Semantic web: define meaning explicitly for objects, represent relationships of domain concepts, enrich annotations, enable users to share and reuse annotations.
- The number of contexts can be any value.
- Rule base: set user's preference with context in annotations.

6.8 Summary

This chapter has presented the development of CAF, a contextual annotation framework for building web-based personalised annotation and personal information systems. The approaches utilises the concepts from Semantic Web, open hypermedia and adaptive hypermedia to provide personalised annotations, support sharing annotations and allow reuse of annotations by storing other annotations in a personal repository. CAF models context in order to provide an adaptable system. Following on from the development the CAF, the next chapter evaluates this approach, using both implementation-based and theoretical studies to demonstrate and critique some of the features of CAF.

Chapter 7 Evaluation

7.1 Introduction

There are two aspects to the evaluation phase, a practical and a theoretical aspect. The research question posed in this thesis is whether or not it is beneficial for annotation systems to use contextual annotation concept and to separate annotations from web pages. The produced annotations should be dynamic in that the contexts of produced contextual annotations objects should match the users' preference. These annotations should be shareable and reusable across applications. The framework should offer all the requirements in Section 6.4. For evaluation purpose, the CAF is instantiated as a prototype, called an Annotation and Personal Information Environment (a-PIE).

The benefit of the CAF approach, to users, is determined by examining the user satisfaction from completing the experiment. The evaluation of user satisfaction asks if the features presented in the a-PIE are useful. This part of the evaluation was carried out by presenting users with a paper-based questionnaire to determine how they used the prototype and their level of satisfaction with different features of the system. The measurements of user's satisfaction used are affect, control, efficiency, helpfulness, Learnability, navigation and comprehension.

The theoretical aspect evaluation of the CAF approach presents in the form of a comparison between this framework and other frameworks or systems that have similar goals and presented in Section 4.5, particularly in Table 4-2. By comparing the a-PIE prototype of the CAF to others, it is possible to draw conclusions as to the advantages and disadvantages of the CAF.

This chapter describes the methods and methodologies used in the evaluation. The user evaluation is presented in the next section while the evaluation of the technical advantages by comparison against other systems is presented later, in Section 7.3.

7.2 The User Satisfaction Evaluation

User studies are necessary to confirm the idea behind the CAF framework in objective and subjective ways. The aim of the evaluation is to examine the following assertion.

The CAF is beneficial to provide adaptable, shareable and reusable annotations.

The 'beneficial' aspects were determined by examining the satisfaction of users on how the CAF approach has impacted their use of the annotations, on the web application. The evaluation of user satisfaction, asks users whether or not the personalised annotations, and features provided by the a-PIE prototype which are creating and viewing contextual annotations, and, personal repository are useful. To test this, an experiment was carried out and presenting users with a questionnaire to determine how they used the system and their level of satisfaction with functionalities of the system.

The hypotheses under investigation were the following.

H_{A1}: Applying the contextual annotation concept in a-PIE increases the speed of navigation.

H_{A0}: Applying the contextual annotation concept in a-PIE does not increase the speed of navigation.

The following section, the experimental design to assess the user satisfaction is presented. It continues with the domain which was used in the experiment, and detail of the experiment. Then the result from the experiment including subjective feedback from questionnaire is shown.

7.2.1 Experimental Design

The experiment was designed to confirm hypotheses stated in the previous section.

User evaluation was aimed at ascertaining the benefit of the annotations with contexts and manipulation of annotations provided by the CAF approach.

The usability criteria used for the evaluation are based on ISO/DIS 9241-1 (effectiveness, efficiency and user's satisfaction) as the followings.

- Effectiveness, such as make sense to users.
- Efficiency, such as maximise user control or speed of navigation.

- User’s satisfaction such as user’s opinion about the system, whether the user likes interacting with the a-PIE prototype. Satisfaction is exhibited by the comfort and positive attitudes users perceive from using the system.

Measurement	Definition
Affect	User’s emotions toward the use of the system.
Control	The degree to which the user feels that they are in control.
Efficiency	The degree to which users can complete tasks in a direct and timely fashion.
Helpfulness	The extent to which the system assists the user in a situation.
Learnability	The degree to which the system is easy for users to learn how to use.
Navigation	The ability for users to move around the system.
Comprehension	The degree to which users can understand the interaction with the system.

Table 7-1 User’s satisfaction measurements (Wills, 2000),(Crowder, *et al.*, 2003)

Within this experiment, each user was introduced to the prototype and had hands-on time with the system to become familiar with it. To provide uniformity in the evaluation process (Wills, 2000), the user was requested to read a written introduction to the a-PIE prototype before conducting evaluation. Paper-based questionnaires were used as a method for information gathering about the prototype. The questionnaire comprised a 5 point Likert-scales questions. At the end of questions there was space for user comments or feedback (see questionnaire in Appendix A User’s Evaluation). Through these areas it was hoped to examine user satisfaction with using the adaptable annotations and functionalities provided by the system. The scales used for measuring user’s opinion or satisfaction was based on the Software Usability Measurement Inventory (SUMI) – *Affect, Control, Efficiency, Helpfulness and Learnability* (Kirakowski and Corbett, 1993), and the scales for evaluating industrial hypermedia – *Navigation* and *Comprehension* (Wills, 2000).

The ability of users to successfully complete their domain tasks in a timely manner means a completion of a task and segments of the task utilizing the information supplied by the user. A task is considered successful if the system is able to comprehend and process the user’s request correctly.

In order to evaluate the CAF approach, the a-PIE system needed to be seeded with annotations with context in order for the users to carry out another session of the experiment. These annotations had been added by the author for authorised group as ‘public’.

To validate the data from the experiments and test for each task, the use of statistical analysis was adopted.

7.2.2 Choosing Suitable Domains for a-PIE

The research would like to ensure that the context object can have more than one criterion. For this evaluation purpose, there were two criteria in one context which are topic and expertise level. The domain topic chosen was about ‘web developing’ as it provided a variety of interesting concepts and each concept could have different expertise levels suitable for users. In the experiment, it was assumed that there were three levels of expertise: beginner, intermediate and advance. The expertise level as ‘beginner’ represented the very simple or easy content, suitable for readers with not much background in this domain. The expertise level as ‘intermediate’ assumed that readers had some background knowledge, and the content contained some technical content. The last expertise level, ‘advance’ assumed that readers were expert in this area, so they needed much background, and the contents would be very technical or needed deep knowledge. Examples of contextual annotation objects used in the evaluation are:

Annotated Area	**Annotation Object**	**Annotation Type**	**Context object**	
			Concept	**Expertise**
In the mid-1960s Ted Nelson coined the word "hypertext" to describe a system of nonsequential links between text.	(some annotations)	Explanation	Date	None
Because of the compression technique used for GIF images this format is best for pictures with spot colors rather than continuous colours.	(some annotations)	Comments	Graphic format	Intermediate
Typography is this relationship between letterforms on a page, playing a dual role in the visual design of the page and in enhancing its readability.	(some annotations)	Example	Font	Intermediate

Table 7-2 Example of contextual annotation objects

7.2.3 Experiment

In the experiment there were four tasks to complete using two systems (see the protocol of the experiment in Appendix A User's Evaluation).

System 1: basic web-based annotation without context and no personal repository. The system provided three functions which were adding annotations, displaying all annotations for a particular URL and displaying all annotations available without any context support. Users could not change the setting.

System 2: web-based annotation with contexts, and personal repository, a-PIE, as a prototype of CAF approach. The system provided three basic functions which were similar to the system 1 except that there was a personalised feature support. It allowed users to add annotations with contexts (Figure 7-1) and select what contexts they wanted annotations to be presented which were set in their preferences (Figure 7-2). In addition users could add interesting annotations into personal repository and also setting personal contexts to the particular annotation. The system also provided personal repository for user to keep interested annotations with personal contexts. List of annotations were given at the top of the particular web page (Figure 7-3).

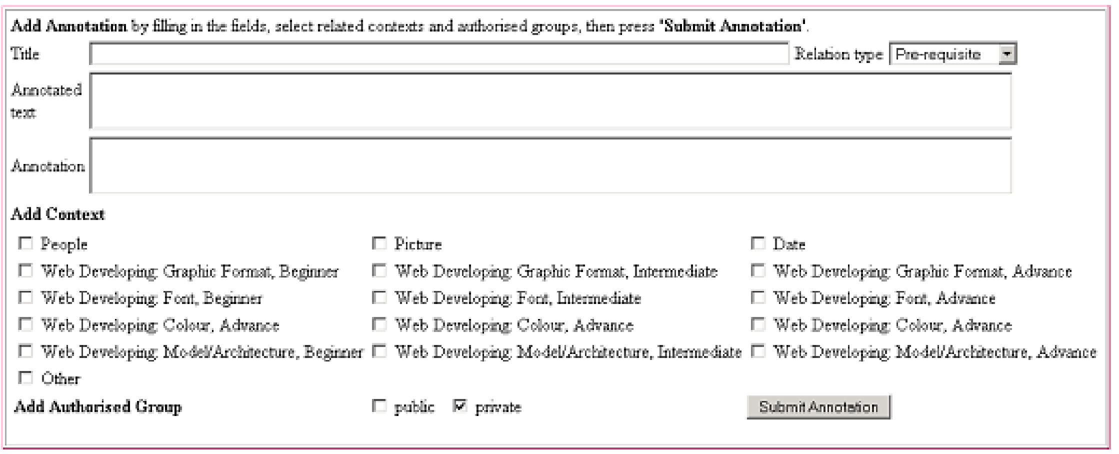


Figure 7-1 Adding annotation with contexts

Figure 7-2 My preferences setting

For simplicity in this experiment the authorised groups were classified as ‘public’ and ‘private’. As the annotations had to be added prior to the experiment, the annotations were authorised for ‘public’ group, which meant that all users were allowed to see these annotations.

The users were asked to perform four tasks during the evaluation. All users had to do all tasks. The description of these tasks is the following.

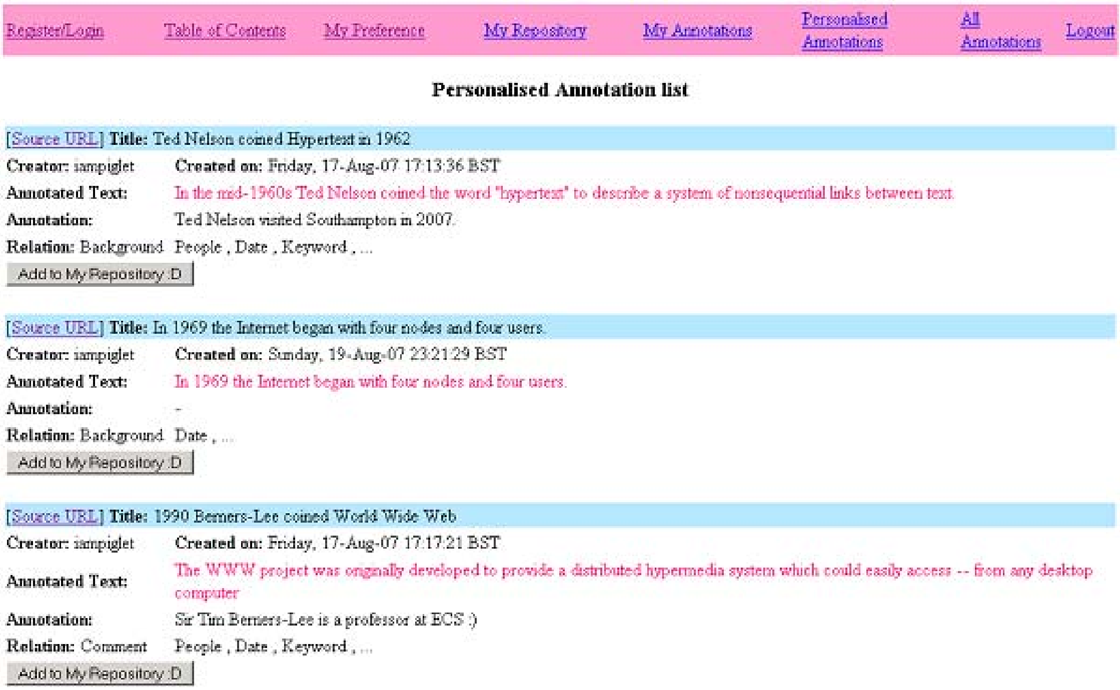


Figure 7-3 Personalised annotation list

Task1 and task 2 were used to test hypothesis H_{A1}; the users had to find answers to a set of questions about the domain (see Table 7-3). Task 1: using the system 1 (annotation system with no contexts) and task 2: using the system 2 (a-PIE). The questions for two groups were different but could get the answer by similar contexts (the difference was only in which system they used to find the answers), and the chosen concepts could be found in the subject domain. The subjects were asked to complete all questions and write down the time taken which was confirmed by the observer.

All users had to do both tasks. For tasks 1 and 2 the group was split into half (A and B), Subgroup A completed the task 1 and then task 2. Subgroup B completed task 2 first and then task 1. This is to remove the learning effect during evaluation.

Task 3: Users had to add annotations to a particular URL and added contexts to the annotations. In order to control the set of annotations for the experiment, the authorisation of these annotations created in this task were set to be the 'private' group. This task was the feature in the a-PIE only. All subjects had to perform this task. They were assigned to add annotations to a part of a URL with specified contexts. The rationale for this experiment was to obtain the users' subjective feedback about the adding annotation with contexts feature of the a-PIE prototype.

Task 1	**Task 2**	**Context**	
		Concept	**Expertise**
Who did coin the word "hypertext"? In which year?	When (in which year) did the Internet begin?	Date	None
What is triadic colour?	What is "split-complementary colour"?	Colour	Beginner
Which graphic format is the most widely supported?	Which graphic format improves image quality and offers better image compression?	Graphic format	Intermediate
Which function delivers the results of processing to the user in a correct, timely, and appropriately formatted fashion?	Which function permits users to interact with the system?	Model/architecture	intermediate
Which figure shows "rasterizing outline fonts"?	Which figure shows "typical core fonts installed on desktop computers"?	Font	Intermediate

Table 7-3 Questions of Task 1 and Task 2

Task 4: Subjects had to add annotations created by others into the personal repository and also set contexts to the annotations. All subjects had to accomplish this task. The aim was to obtain the users' subjective feedback about the repository feature of the a-PIE prototype.

A questionnaire was given to the users after completing all tasks (see Appendix A User's Evaluation).

The summary of experiments and the features of each are given in Table 7-4.

Experiment	Features
Task 1 and 2: query annotations with contexts and with no contexts	querying annotations (with/without contexts for a URL or all URLs in the web site.) shareable of annotations adaptable of annotations
Task 3: add annotations with contexts	adding annotations
Task 4: add annotations with contexts into repository	reusing annotations

Table 7-4 Tasks and Features of Each Task

7.2.4 Description of Subjects

The subjects taking part in the experiment were from voluntary postgraduate students and researchers at the University of Southampton. They had various education backgrounds. One subject was a post doctoral researcher, 8 subjects were master students and the rest were PhD students. Table 7-5 shows the distribution of subjects in background and gender.

Education Background	**Gender**			
	Female	Male	**Total**	**%**
Computer Science	4	4	8	31
Education	1	0	1	4
Engineering	3	4	7	27
Finance & Management	3	2	5	19
Other Sciences	3	2	5	19
Total	14	12	26	100
%	53.85	46.15	100	

Table 7-5 The distribution of subjects

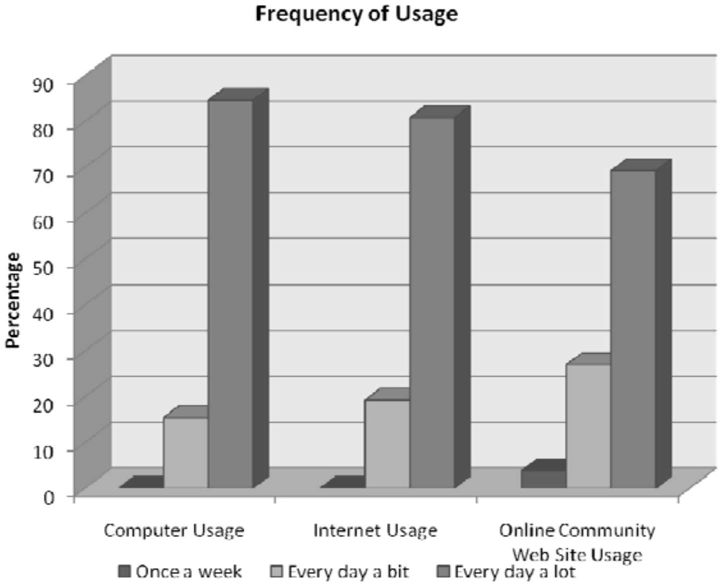


Figure 7-4 The distribution of subjects in terms of computer, internet and online community Web site usage

As may be seen in Figure 7-4 all of subjects used computers everyday. Most of them, 85%, used computers ‘very much’, while some of them, 15%, used computers ‘little’. That means they were familiar with using the computers. It also shows that most of them, 81%, accessed the internet much each day while some of them, 19%, used the internet ‘little’. More than half of them, 69%, use online community web sites a lot while 27% use a little bit each day. The purposes of using online community web sites were to share or exchange idea with others, relax, communication, and seek for information.

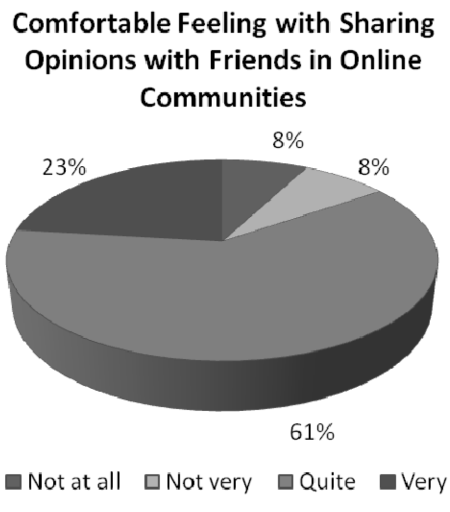


Figure 7-5 Level of comfort with sharing opinions in online communities

As shown in Figure 7-5, some subjects, 23% felt very comfortable to share opinions in online communities, while 61% feel comfortable. A few subjects, 8%, did not want to share and 8% not very liked to share information in online communities. However, majority of subjects who very often use online community web sites, 42%, feel quite comfortable to share opinions with others, while 19% of subjects feel very comfortable to do that.

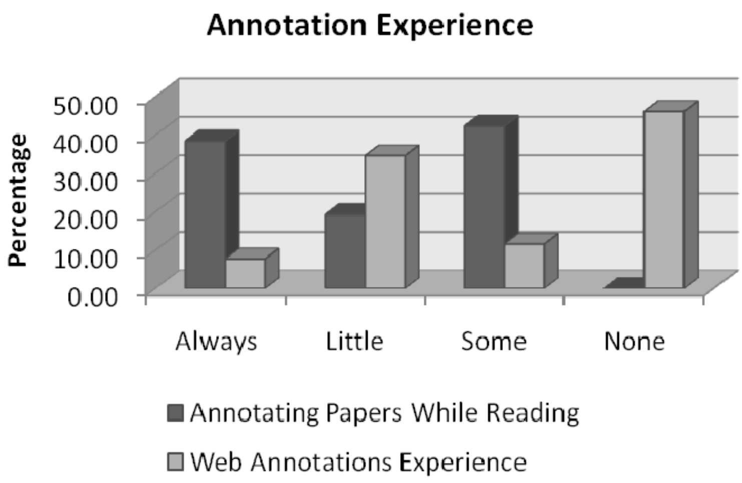


Figure 7-6 Experience of annotation

Figure 7-6 presents annotation experience of subjects. All of subjects made annotations while they were reading papers. There were 42% of subjects who made annotation on papers sometime when they read papers, while 39% always made annotations while reading and 19% of subjects seldom made annotations. Based on web annotation experience, only one subject always used social bookmarking systems such as del.icio.us, while nearly half of subjects never used web annotations. The rest of subjects claimed that they often, sometime or always used bookmark from the browsers.

The purposes of making annotations, from subjects' opinions, were to highlight important information or main ideas, remember information, refer back, note or reminder for future or to note the idea to reuse it next time. In addition, annotations could point out the main topic, important parameters, some interesting numbers/contexts which made it easy to find or remind later.

Discussions

These initial questions about subjects might impact to the evaluation. The goal of capturing this information was to determine the subjects' level of computer skills, level of comfort with using online community Web sites. It was felt that this was an important metric which to evaluate the subjects' satisfaction with the features of the a-PIE prototype. For example, if the subjects' stated that they were uncomfortable with sharing opinions with others in online community then this may be used as a context in which to view their answers to questions about their use of the annotation sharing and reuse.

As all subjects were studying or working in the university environments, they were familiar with the computers, the Internet and online community web sites. Since the a-PIE was a web-based application, the subject's ease in using a computer, the internet and online community web sites were important considerations. In addition, more than half of them were willing to share information within online communities. They understood purposes of making annotations. Most of them made annotations while they were reading on papers, while some of them used bookmarking integrated with browsers. Education background did not relate to the Internet usage or even community web sites usage.

The decision to base the domain content in the subject area of 'web developing' seems justified as there were diverse education backgrounds among subjects. If the subjects using the adaptable annotation all had the same previous background then the personalisation may have been less effective.

7.2.5 Efficiency of the CAF Approach for Annotations

To examine the efficiency of the annotations with contexts concept as applied in the web-based annotation system in comparison with the basic annotation system without the presence of personalised feature, the users were asked to answer the questions using two different systems; system 1 and system 2, and write down the start and finish time which was confirmed by the observer.

Hypothesis:

H_{A1}: Applying the contextual annotation concept in a-PIE increases the speed of navigation.

H_{A0}: Applying the contextual annotation concept in a-PIE does not increase the speed of navigation.

Independent variable: system 1 (the basic annotation system), and system 2 (a-PIE).

Dependent variable: Time usage to finish the tasks and percentage of correctness of answers of each task.

Results for the experiment

In this experiment, the time usage to finish the tasks and percentage of correctness of answers were used to define the speed of navigation (the less time usage to complete the questions, the higher speed of navigation).

All subjects answered all questions correctly with different times. Therefore only time was used in the analysis. The time usage of users from two systems is given in Figure 7-7.

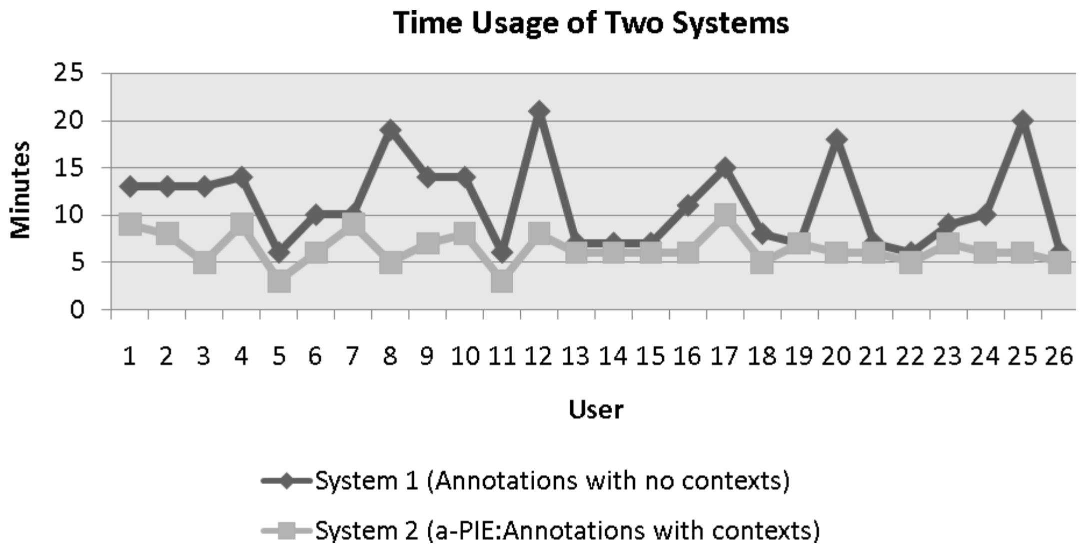


Figure 7-7 Time Usage of Two Systems from questions in Table 7-3

Time usage	Mean	N	Std. Deviation	Std. Error Mean
System 1	11.19	26	4.665	.915
System 2	6.42	26	1.748	.343

Table 7-6 Paired sample statistics for the system1 and 2 for Task 1 and Task 2 produced by SPSS

	Paired Differences					t	df	Sig. (2-tailed)
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower	Upper			
System 1 - System 2	4.769	4.236	.831	3.058	6.480	5.741	25	.000

Table 7-7 Result of paired samples test produced by SPSS for the paired sample t-test for Task 1 and Task 2

Paired sample t-test was used to test for significant differences between two groups from data that has been gathered by way of a related measures design (Kerr, *et al.*, 2002). According to Table 7-6, time usage (minimum = 3, maximum = 10, mean = 6.42 minutes) of the system 2 (the a-PIE system), was less than of the system 1 (the basic annotation system) (minimum = 6, maximum = 21, mean = 11.19 minutes). The difference of time usage supported the hypothesis that the annotation with contexts concept applied in a-PIE significantly increased the speed of navigation at the 5% level of significance ($t = 5.741$, $df = 25$, $p < 0.05$), as given in Table 7-7.

Generally it is believed that the annotation with contexts concept has impacted the speed of navigation. The web-based system that applied the personalised feature by supporting the annotation with contexts concept and allowing users to choose annotations based on their preferences significantly increases speed of navigation.

7.2.6 Results of tasks 3 and 4.

In this section the results for each of the scales from the questionnaire are presented.

Control of the System

The questions asked in this area were to capture the users' experiences of the features of the system 2 (a-PIE), controlling the system and ask for opinions of the annotation with contexts concept. It examined the beneficial of annotation with contexts concept serving by the ontology-based contextual annotation service in the CAF as applied in a-PIE.

Figure 7-8 shows summary graph of the users' feedback concerning the control's statements. In order to manipulate annotations such as when creating annotations or querying annotations approximately 80% of subjects agreed that the system should have some predefined contexts for users to choose in 'personal preferences' function, while 3.8% (one user) did not like it and 15.4% (four users) neither agreed nor disagreed (Question 5).

Here all of users 100% would like to create their own preferences apart from the predefined contexts provided by the system (Question 6).

Most of the users (96.2%) liked that they could set personal preferences in order to get personalised annotations. None of them disagreed while only 3.8% (one user) neither agreed nor disagreed and none of them disagreed (Question 7).

To examine the degree to which the users felt that they were in control, in other words the annotations reflected what they specified in the 'personal preferences', a majority of users approximately 88% agreed while 11.5% (three users) neither agreed nor disagreed and none of them disagreed (Question 8).

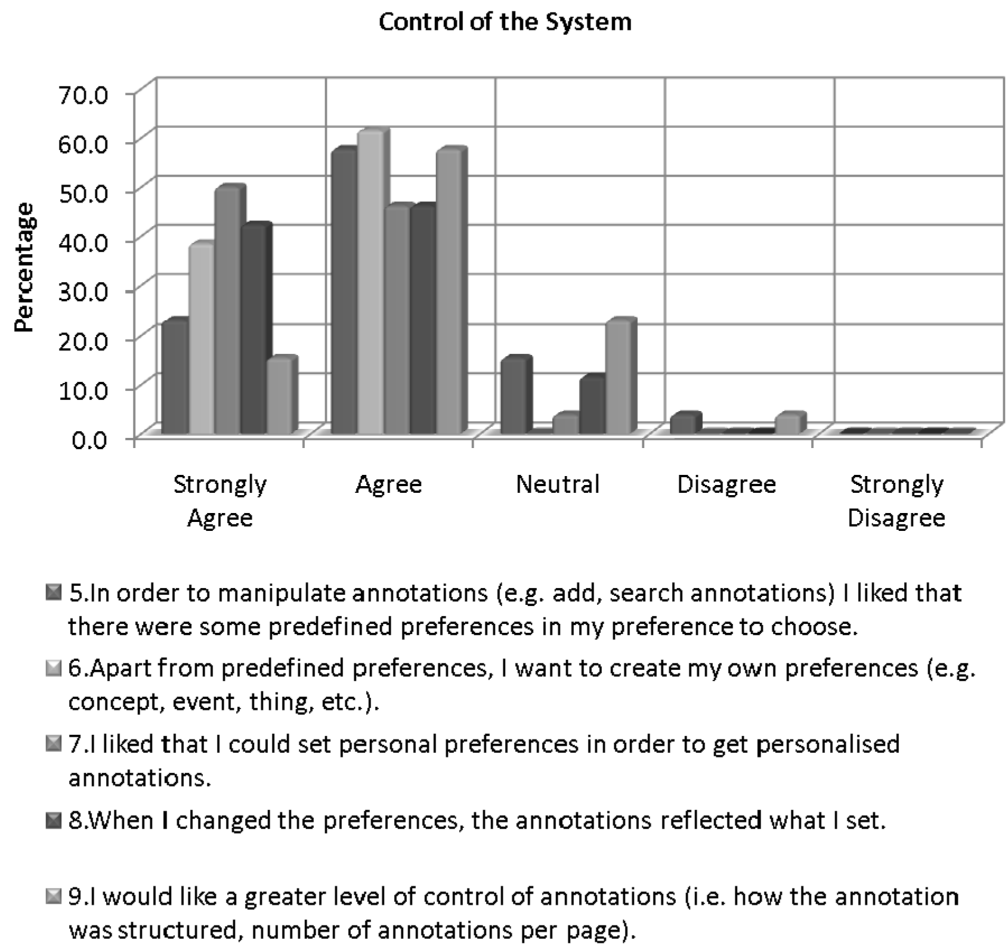


Figure 7-8 Subjective feedback on the 'control' aspect of the System

As the system delivered all annotations according to contexts set in 'personal preferences' sometimes users felt that they would like a greater level of control of these annotations such as how the annotations were structured or number of annotations per page. Most of users (72%) agreed that they wanted more control, while 3.8% (one user) disagreed and 23.1% neither agreed nor disagreed (Question 9).

The results from these questions in this section indicates that most of users preferred to have predefined contexts in 'personal preferences' and also wanted more flexibility to create their own contexts in 'personal preferences' for personal purposes. However, a few users were satisfied with what the system provided and did not like the complicated setting but preferred the simple one.

When users felt that they were in control the system, it indicates that the a-PIE system could perform and deliver annotations correctly according to users' preferences.

Efficiency

These statements of the users' opinion questionnaire measured the degree to which users could complete tasks in a direct and timely fashion. Figure 7-9 illustrates a summary graph of the users' opinions concerning on the efficiency's statements.

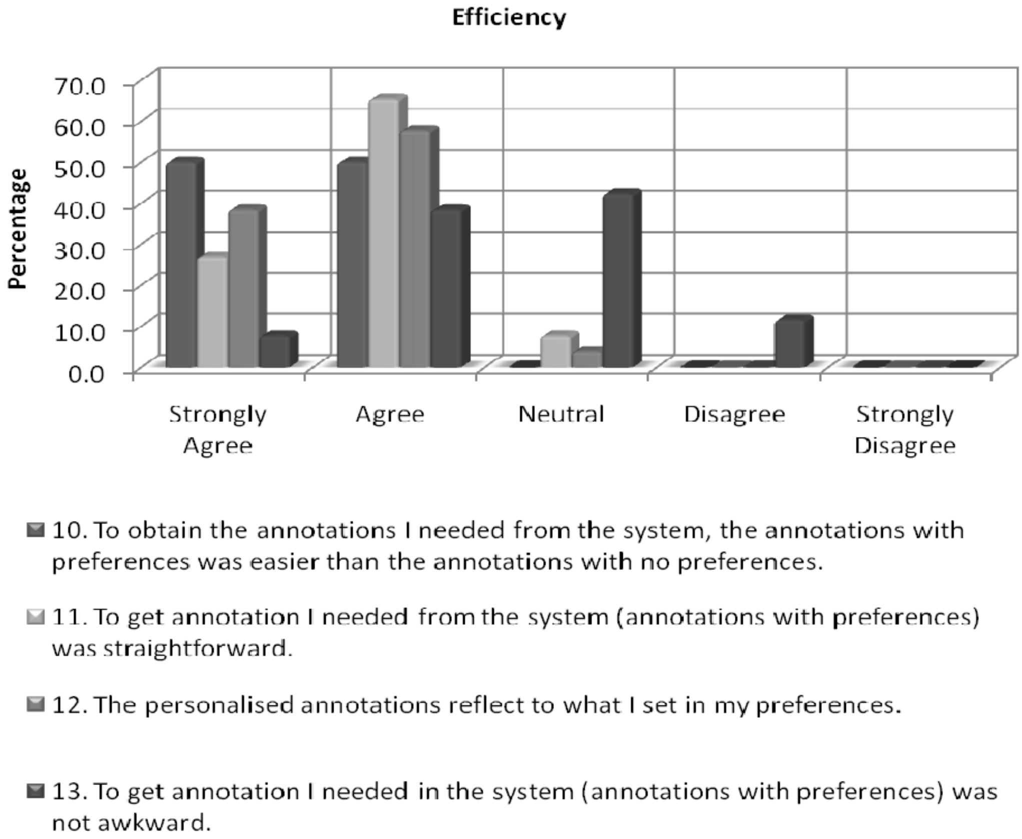


Figure 7-9 Subjective feedback on the 'efficiency' aspect of the system

As can be seen in Figure 7-9, all users agreed that to obtain the annotations for a particular URL including annotations in 'my annotations' and 'my repository' from the system with the annotation with contexts concept was easier than the annotations with no preferences setting (Question 10). This indicates that the system with annotation with contexts was significantly easy to use and it also increased the speed of navigation.

There were 92% of users agreed that the process to get annotation was straightforward, while only 7.7% (two users) neither agreed nor disagreed and none of respondents disagreed (Question 11). In addition, only one user (3.8%) neither agreed nor disagreed that annotations produced from the system reflected to what they set in their preferences, while approximately 95% of users agreed (Question 12). This could indicate that the system was consistency and worked properly.

Nearly 50% of users agreed that getting annotations was not awkward while 43% neither agreed nor disagreed and 12% disagreed (Question 13). This indicates that the processes of getting the annotations provided by the system was troublesome for some users. However, the objective of this prototype of the CAF approach was concentrate on the concepts of the CAF approach using the annotation with contexts, not the user interface. The result of this statement was useful and the user interface could be improved or extended as a future work. The improvement or extension of user interface is discussed in Section 7.4.

Helpfulness

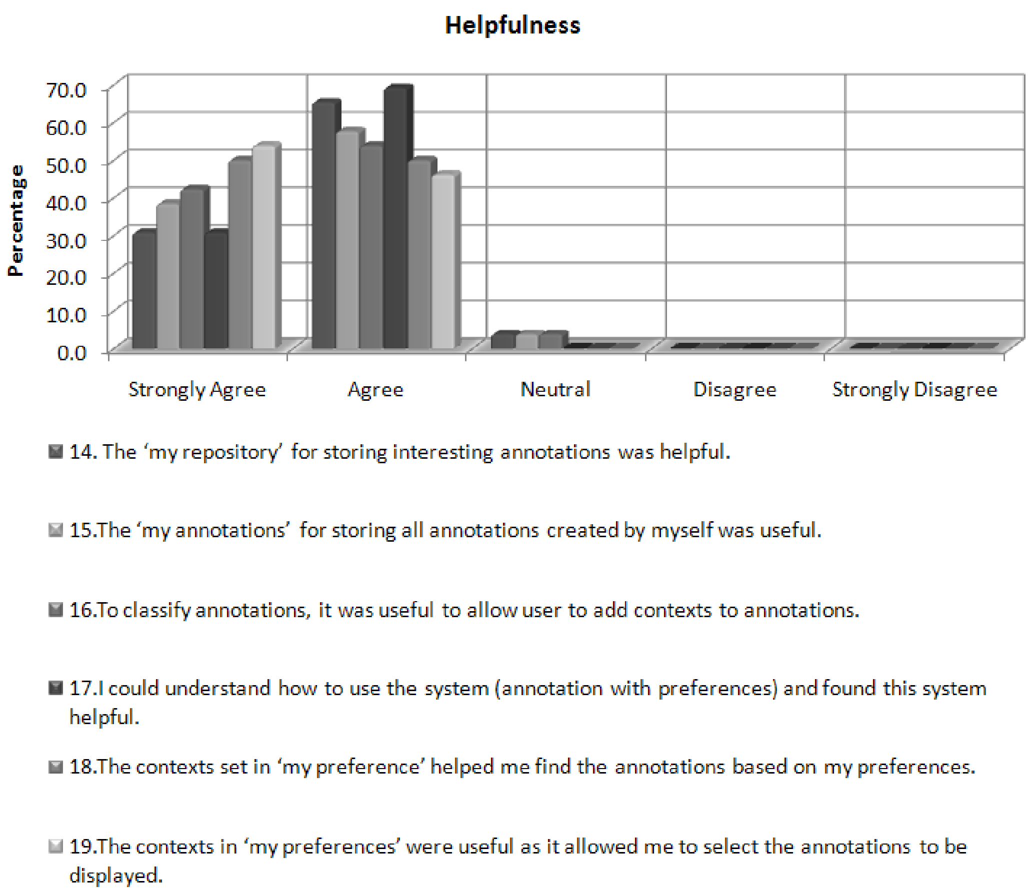


Figure 7-10 Subjective feedback on the 'helpfulness' aspect of the system

Six statements were set to assess whether the system assisted the user in a situation. The features provided in the a-PIE prototype were 'my repository' for storing interesting annotations, 'my annotations' for storing all annotations created by themselves and 'my preferences' for specifying which contexts of annotations to be displayed. This includes the helpful of the 'annotation with contexts' concept to classify annotations and searching annotations.

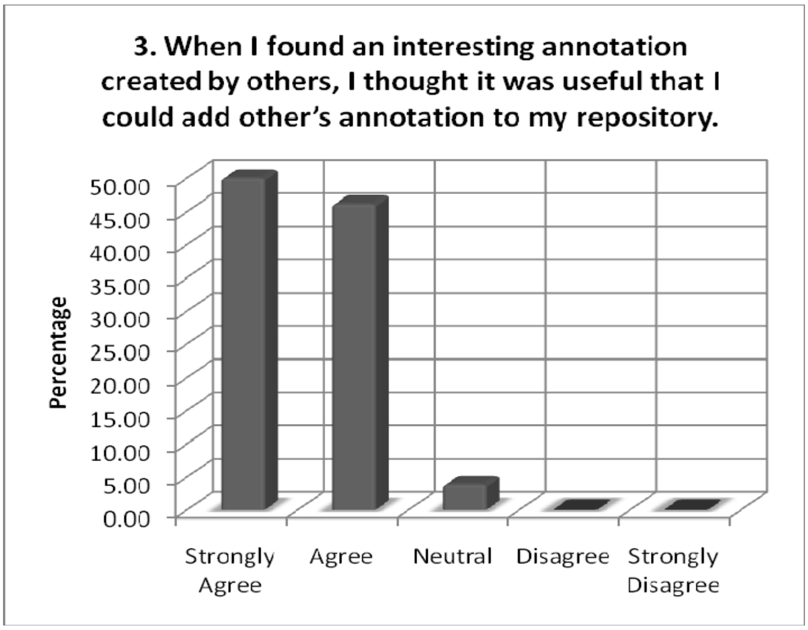


Figure 7-11 Subjective opinion about 'my repository' feature

As can be seen in Figure 7-10, a majority of the users 96% agreed that 'my repository' was helpful, whereas none of the participants disagree and only 3.8% (one user) neither agreed nor disagreed (Question 14). Similarly to the 'my annotations' function, only one user (3.8%) neither agreed nor disagreed while the rest 96% agreed that this function was useful (Question 15). To classify the annotations, only one user (3.8%) neither agreed nor disagreed while the rest agreed that it was useful to allow user to add contexts to annotations (Question 16). All users agreed that they understood how to use the system and found this system helpful (Question 17). All of users also found that contexts set in 'my preferences' helped them find the annotations based on their preferences (Question 18) and the contexts set in 'my preferences' were useful as it allowed them to select the annotations to be displayed (Question 19).

Figure 7-11 shows that only one user (3.8%) neither agreed nor disagreed while the rest agreed that 'my repository' feature was useful.

This result indicates that these features provided by the CAF approach as applied in the a-PIE was significantly useful and helpful.

Learnability

To examine the degree to which the system was easy for users to learn and use three statements were presented.

Figure 7-12 shows a summary of the users' opinions about the learnability of the system. As can be seen in the figure, a majority of users 88.5% agreed, none of participants disagreed and strongly disagreed while 11.5% (three users) neither agreed nor disagreed that the learning to use the system was straightforward (Question 20).

The system was easy to learn and use with 78% agreed, while 15.4% (four users) neither agreed nor disagreed and none of users disagreed (Question 21).

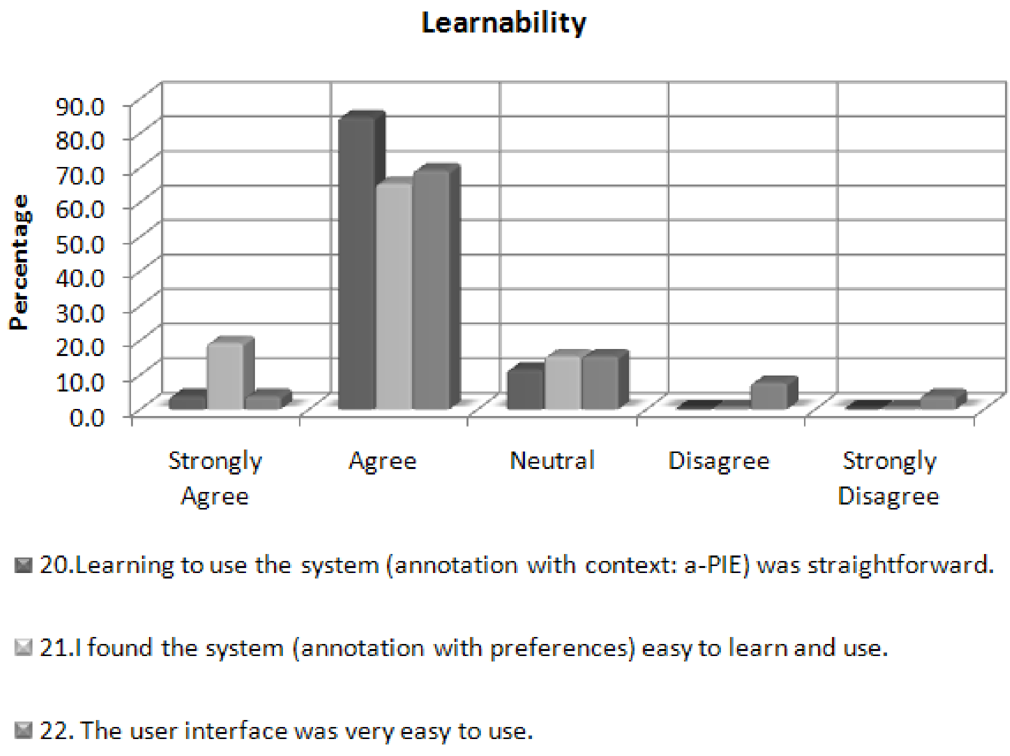


Figure 7-12 Subjective feedback on the 'learnability' aspect of the system

Majority of users, 74% of participants agreed that the user interface was very easy to use. However, 15% neither agreed nor disagreed and 11% disagreed (Question 22). These four users who neither agreed nor disagreed also neither agreed nor disagreed that the process of getting the annotations was awkward (statement 13 as shown in Figure 7-9). One of two users who felt that the user interface was not very easy to use felt that the process of obtain the annotations was awkward, whereas another neither agreed nor disagreed. This indicate that the user interface might not be very easy to use for some users however it was straightforward to learn and users could get use to it if they were given more time. The improvement of user interface is discussed in Section 7.4.

Navigation

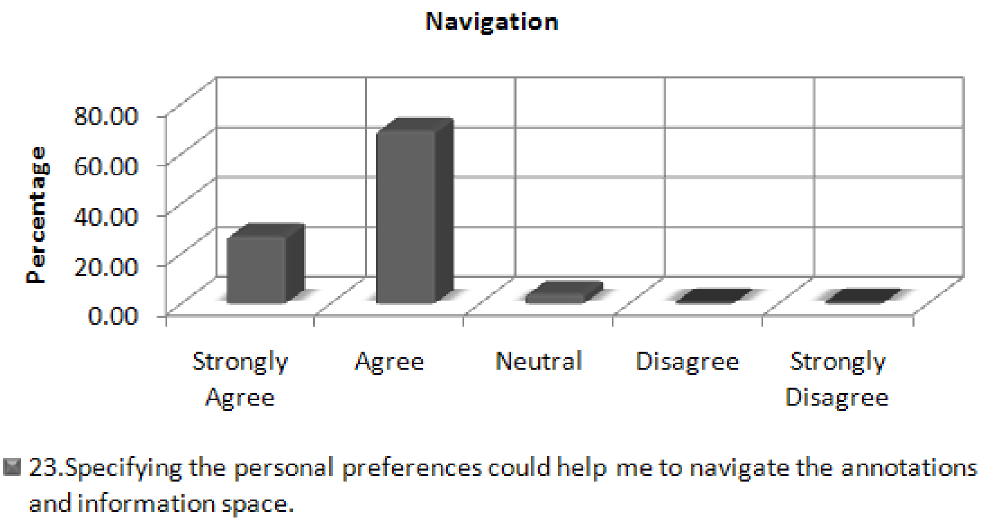


Figure 7-13 Subjective feedback on the 'navigation' aspect of the system

Figure 7-13 illustrates the ability for users to move around the system. As can be seen in the figure, majority of users 96% agreed that the specifying the personal preferences could help them to navigate the annotations, while none of participants disagreed and 3.8% (one user) neither disagreed nor agreed. This indicates that the system with the personal preferences supported the navigation of the system.

Comprehension

Four statements were designed to assess the degree to which users could understand these interaction with the system. The summary of the feedback is shown in Figure 7-14.

As can be seen in Figure 7-14, a significant majority of the users 82% understood the interaction with the system whereas 12% neither agreed nor disagreed and none of them disagreed (Question 24). Similarly, most of users 84% felt that the system was clearly presented and understandable and while 15.4% neither agreed nor disagreed, and none of them disagreed (Question 25).

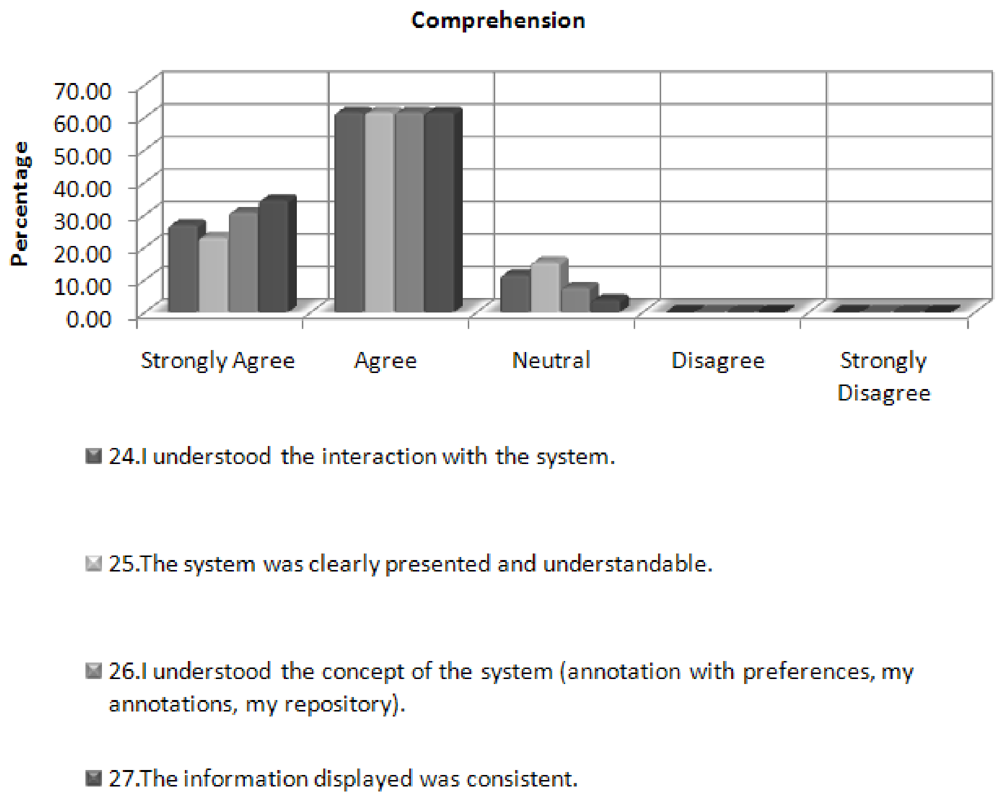


Figure 7-14 Subjective feedback on the 'comprehension' aspect of the system

The users understood the concept of the system with a significantly majority 91% agreed while 7.7% (two users) neither agreed nor disagreed and none of them disagreed (Question 26). A significant majority 96% of users felt that the information displayed was consistent while 3.8% (one user) neither agreed nor disagreed, and none of them disagreed (Question 27).

This information indicates that these results from the evaluation and questionnaire were reliable in that a significant majority of users understood the interaction with the system and they knew the concept of the system presented. In addition, it indicates that the system functioned correctly.

Affect and Overall Satisfaction of the CAF approach

Three statements measured the users' emotions toward the usage of the prototype system. Figure 7-15 shows a summary of the participants' opinions on the 'affect' aspect.

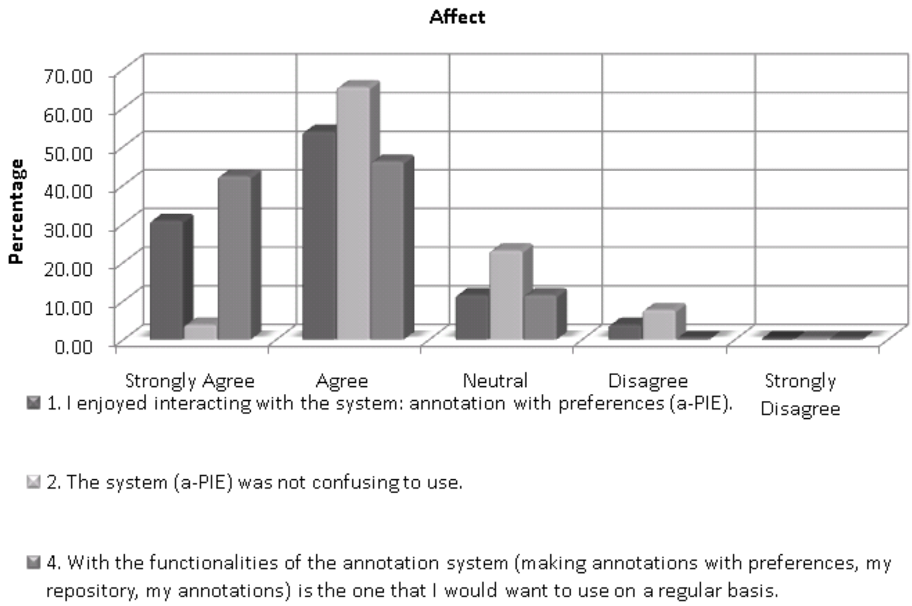


Figure 7-15 Subjective feedback on the ‘affect’ aspect of the system

As can be seen in Figure 7-15 a significant majority of users 84% agreed that they enjoyed interacting with the a-PIE system, while 11.54% neither agreed nor disagreed and 3.85% disagreed (Question 1). Nearly 70% agreed that the system was not confusing to use while 23% neither agreed nor disagreed and 7% disagreed (Question 2). A significant majority of users 88% wanted to use the system on a regular basis while 12% neither agreed nor disagreed (Question 4).

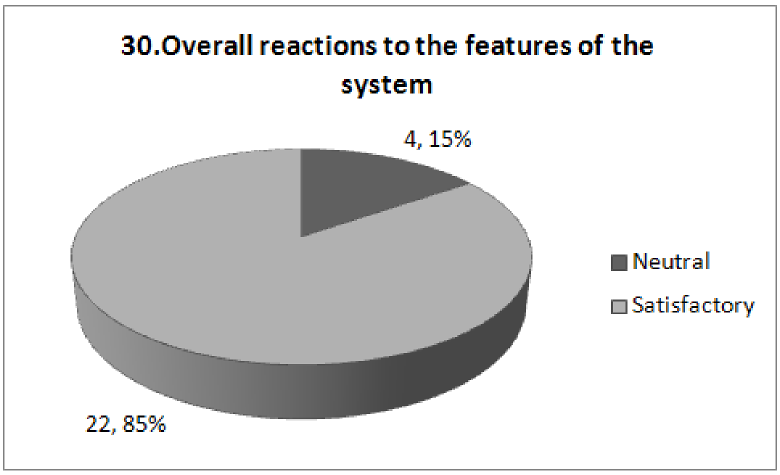


Figure 7-16 subjective feedback on the overall reactions to the features of the system

The information in this section, as illustrated in Figure 7-16, is asking users about overall reactions to the features of the system. It shows that a significant majority of users with 85% satisfied the features of the system while 15% neither agreed nor disagreed.

User Feedback

This user evaluation provided an opportunity to obtain user feedback on the CAF approach which applied annotation with contexts concept in the a-PIE prototype. All participants in the experiments were first time users. At the end of the experiment, the user indicated that the user interface was not very good. However, this prototype was not concentrated on the user interface but the concept of the CAF approach.

As some users were familiar with this domain, they thought that the contexts shown in the system was adequate, while some thought that the expertise level might not be needed. Some users agreed to have knowledge level as a context, but some thought that is was not necessary. Some users felt that the complicated contexts require expert to do, therefore the simple context (only one dimension such as concept) was enough. However, some users felt that one context with many conditions was also useful in that it allowed users to classify or categorise information. A user mentioned that it was better to have standard for contexts in order to share them with others in a community.

The language might be more useful as a dimension of a context in other circumstance. It would be more useful if the users could type or read other users' annotations in his/her language. However, they concerned that this would make 'my preferences' function more complicated.

Some users gave feedback on additional features of the system that are summarised here.

- It would be great if the user interface was improved and the users could make annotations by highlight with different colours.
- It would be great if we could annotate pictures on the web.
- One user preferred annotations that were annotated automatically by the system.
- The system might show the reputation of users who created useful annotations.

User comments on the concept, framework and features of the system can be summarised as follows:

Users thought that the contextual annotation concept in a-PIE was an interesting concept and all of them thought that it was useful to access information on the Web site that allowed users to annotate information and it was useful to remind people to remember these annotations later. Comparing the annotations with contexts and with no contexts, the annotations with contexts were more useful and required less time to find annotations than the annotations with no contexts. Users thought that sharing these annotations and also keeping annotations created by others were useful for other people who also were interested in the same annotations and wanted to share their ideas with others. And lastly, the concept and prototype are practical and easy to use.

7.2.7 Summary from the User Evaluation

The result from user evaluation has shown that the CAF approach which was prototyped as the a-PIE was effective in that the system could deliver annotations according to the users' preferences. In addition these annotations made sense to users. The result in the control aspect of the system could indicate the significant of the effectiveness in that it was reasonable to users.

The CAF approach applying the annotation with contexts is beneficial in providing dynamic annotations for online community web sites. This confirmed from the experiment in task 1 and task 2 and subjective feedback from questionnaire. The time usage to answer questions was significant less in the a-PIE (annotations with contexts) than in the system 1 (annotations with no contexts). This indicates the efficiency of the system. A significant majority of users agreed that they had control the system as shown in Figure 7-9.

The opinions of users on the context concept were that most of users agree that annotation with contexts concept was useful. The context could help classify annotations and personalise annotations. Most users preferred predefined contexts, and also would like to create their own contexts. However, the specification standard of context was another interesting issue. A few users commented that context consisting of many conditions was a complicated task which needed expert to do.

The CAF approach is beneficial in providing shareable annotations. The annotations for users in the system 2 (a-PIE implemented from the CAF approach) were created and intended to share to 'public' group. As in this experiment, every user was assigned in the ‘public’ group, therefore when these users could see annotations this indicates that the framework supported the sharing of annotations. In addition, there were comments from users that sharing of annotations was useful, reasonable and helpful in that we could see others’ views or opinions.

The CAF approach is beneficial in providing reusable annotations. This is the ‘my repository’ feature of the a-PIE prototype. The result of the user evaluation showed that this feature was significantly useful, as shown in Figure 7-10 (statement 15) and Figure 7-11. Comments from users were that it was useful that users could save other annotations into personal repository because sometime others opinions were interesting and we did not need to create a duplicated annotation. Users could set personal context to the annotation which was created by other, before saving it into ‘personal repository’ was useful and reasonable in that sometime we may interested in those texts in different points from others

Overall satisfaction with the usability of the features was significantly high amongst the users. It is shown that users significant preferred the adaptable annotations from result of statement 16, 18, 19, as shown in Figure 7-10. Users wanted to set the contexts to annotations in order to describe or classify the annotations they made. This is significantly confirmed from the result of statement 5, 6, 7, as shown in Figure 7-8 and Figure 7-10.

7.3 Comparisons with related systems

The purpose of this section is to compare the presented framework with systems that share some common features and goals. The system is compared with some web-based annotation systems presented in Section 4.5.1 and Section 4.5.2.

Table 7-8 illustrates the classification of annotation systems excerpted from Azouaou’s categories. There are two factors and four properties. Two factors are the annotator and the user of the annotation. These three factors provided four properties of annotation tools. First, manual annotation means that all three functions of annotations are performed by a human. There are three functions of annotation activity which are selecting an annotated document (or region) as a source, selecting an element as a destination and defining the properties of relation itself. Opposite to manual annotation, automatic annotation means that these three functions are performed automatically by a software agent and semi-automatic annotation means that at least one of these three processes are assisted by the software tools. Second, there are two properties describing the annotation audience; cognitive and non cognitive aspects. Cognitive aspect of annotations represents an annotation in a visible shape. Finally, if the annotation is supposed to be used by a software agent then it is a computational aspect which is opposite to non computational aspect (Azouaou, *et al.*, 2004).

Addressee/Author	**Manual**	**Semi-automatic**	**Automatic**
Cognitive and non computational annotation	Annotea, AWF, Yawas	**CAF**	
Non cognitive and computational annotation	SHOE		
Cognitive and computational annotation		CREAM	Magpie, COHSE

Table 7-8 Classifying annotation systems using Azouaou's categories (Azouaou, *et al.*, 2004)

Based on the semantic annotation properties given by Azouaou, some semantic annotations, presented in Section 4.5.1 and Section 4.5.2, were classified in Table 7-8. Annotations in Annotea, AWF, Yawas and CAF were manipulated by users. While Annotea, AWF and Yawas provided all annotations manipulated by users, CAF did not. CAF should be classified as a semi-automatic annotation tool for the following reasons. In CAF the contextual annotation objects would be delivered only a list of annotations that were dependent on contexts chosen by users. The human agent is helped by the software tools to perform at least one of the three annotation sub-process. In addition, there were semantic defined explicitly to the annotations. Table 7-8 illustrates how CAF fits into the Azouaou's semantic annotation system categories.

Considering the requirements given for the scenario in Section 1.2, CAF fulfils all the requirements as presented in Table 7-9. CAF supports the manipulation of annotations. CAF enables users to set authorisation for each annotations while it allows users to add others' annotations and set their own contexts into personal repository which is considered as reusability. The CAF support the personalisation in that it produces contextual annotation objects based on user's preferences. In addition, CAF separates annotations from web content and define contextual annotation structure formally in

RDF(S)/OWL as semantic web languages, which assure that it offers semantic interoperability and syntactic interoperability as specified requirements in Section 6.4).

Requirements / Systems	Annotation	Shareability	Personal repository (Reusability)	Annotations with context	Personalisation	Annotation Storage (Reusability)	Formats (Syntactic and Semantic Interoperability)
CAF	✓	Public, private, group	✓	✓	✓	Annotation Server	✓ (RDFS, OWL)
ComMentor	✓	Public, group, private	X	X	✓ (trail, tour)	Annotation server	X (Text)
Yawas	✓	Private (local file), public	X	X	X	Text file	X (HTML, DOM)
CritLink	✓	Public	X	X	X	Annotation server	X (HTML)
Annotea	✓	Public	X	X	X	Local or annotation server	RDF(S), XLink, XPointer
CREAM	✓	N/A	X	X	X	N/A	RDF, OWL, XPointer
AWF	✓	Private, share	X	X	X	AWF server	✓ (RDF)
SHOE	✓	N/A	X	X	X	Embedded in Web page	X (SHOE)
COHSE	✓	N/A	X	X	X	Annotation server, DLS	DAML+OIL
WiCK	✓	N/A	X	X	X	Annotation server (3store)	X (MS smart documents)
Magpie	✓	N/A	X	X	X	None, real-time	X (HTML, OCML)

Table 7-9 Summary of various properties of the annotation systems or frameworks

7.4 Discussion

While there are some annotation systems or frameworks available, none of them support the annotation with context. The context object attached to the annotation object is an important object that systems can use to provide personalised or adaptable annotations instead of delivering all annotations to everyone. It is worth mentioning that the CAF proposed can provide adaptable annotations.

The decision to use RDFS/OWL to model the ontology used in CAF has advantages in that it provides semantic and syntactic interoperability. This supports the shareability and reusability of annotations in the same systems or across systems. To implement the CAF is not very complicated as there are several RDFS/OWL parsers or toolkits that can handle the ontology. Each part of structure in CAF can be extended by referencing through the URI. For example, the context object can have several criteria. Each criterion is combined through the URI of a context object.

Separating the contextual annotations from web content, using the link service approach enables the independent manipulation of annotations. When there are changes in the annotations there is no effect to original web content. The ontology-based contextual annotation service that is implemented to manipulate the contextual annotation objects functions like the link service.

Each contextual annotation structure, in CAF, is more complicated than implementing it in RDF. Structuring contextual annotations by extending the FOHM ontology has advantages in that it can support several structures such as navigation link or tour. It supports adaptability, shareability and reuseability at every part of the structure. A Reference object can be used to specify the location of an annotated region. The content of an annotation is kept in a Data object. While the context in which the annotation is make is stored in the Context object which can be attached to any other object and is used by the personalised mechanism.

Designing CAF as a web services architecture supports shareability and reusability. Each service serves their own function such as the user service for manipulate user information and preferences, while the context service handles the context value. A change to any service has no effect on the other services.

The studies were carried out to reflect the principle of evaluation of CAF. The user evaluation was conducted to examine the user satisfaction of the features provided by the CAF. Apart from the user interface, users are agreed that CAF is useful for annotating web sites and for learning, as they can learn from others' annotations. In addition, users are satisfied with the features of CAF. These features are personal repository, the contextual annotation and sharing contextual annotations by authorisation. In the current version of the prototype the user interface is not very easy to use as all annotations are displayed separately as a list on every web page, without highlighting applied to the web content. However, the user interface can be improved by highlighting annotated regions in different colours for different annotation types, for example, red for comment or blue for explanation. Web technologies like AJAX can be used to implement highlighting by augmenting the annotation on the web content.

7.5 Summary

This chapter has presented the evaluation that was conducted with the objective of examining the beneficial and theoretical aspect of the CAF approach which applied the annotation with contexts concept in the a-PIE prototype. The data was gathered objectively and subjectively, and analysed by means of statistical techniques. The theoretical aspect was assessed through the technical advantages and disadvantages compared against other systems. The results of the user evaluation showed a significant level of acceptance from the ‘useful’ and ‘satisfy’ aspect of personalised annotation with shareability and reusability provided by the CAF approach.

The following chapter provides a conclusion of the research undertaken in this thesis. The contributions this work has made as well as key research issues raised are particularly discussed. The chapter then concludes with the highlights of the possible directions this work could be continued.

Chapter 8 Conclusion and Future Work

This final chapter provides a summary of the work presented in this thesis. The proposed contextual annotation concept and the contextual annotation framework are presented, followed by the evaluation undertaken. Then the chapter concludes and highlights the possible directions for future work.

8.1 Summary and Conclusions

8.1.1 Contextual annotation framework

This thesis has proposed the concept of contextual annotation framework (CAF) for web-based annotation systems. The CAF framework attempted to support the adaptable of annotations and shareability and reusability of annotations as research motivation presented in Section 1.2. The concept of contextual annotation proposed means that each annotation object should have context objects to identify criteria or conditions that each contextual annotation object will be delivered based on an individual user's preference. The context object can consist of several conditions, for example topic and expertise level. The CAF approach provides adaptable annotation by using contextual annotations, in that the context object can be used in modelling decisions as part of the personalised mechanism.

The ontology used in CAF is defined formally in RDF(S)/OWL in order to support the shareability, and, syntactic and semantic interoperability. By implementing ontology in RDF(S)/OWL, each part of contextual annotation, user model, repository model and structure model can be shared and reused through the ontology. As RDFS/OWL is a semantic web language the CAF approach ensures that the semantics of contextual annotation can be shared across systems.

Contextual annotation objects are manipulated separately from web content through the ontology-based contextual annotation service. The ontology-based contextual annotation service is designed and implemented based on the link service approach (as described in Section 3.6), therefore the contextual annotation objects are transferable and also manipulated; add, edit, delete independently.

Contextual annotation object ontology is designed by extending the FOHM ontology (Gibbins, *et al.*, 2003). While FOHM (represented in Section 3.5) is a model which enables context objects to be attached to any part of FOHM structure, it does not fit the requirements of CAF for the syntactic and semantic interoperability. CAF extended the FOHM ontology (Gibbins, *et al.*, 2003) which is based on the RDF/OWL. Therefore the contextual annotation object ontology supports the syntactic and semantic interoperability which are requirements of the CAF approach.

The CAF fits all of the requirements specified as the research motivation of this thesis (presented in Section 1.2), while other systems do not, as given in Table 7-9.

8.1.2 Implementation

The work has presented the application of the CAF in a prototype, an annotation and personal information environment (a-PIE). The a-PIE is a web-based personalised system. The ‘web developing’ domain was used in the prototype. An ontology of storage models has been designed, represented in RDF(S)/OWL, and several services have been developed. In the prototype the context object consisted of two dimensions which are ‘concept’ and ‘expertise level’. The prototype was implemented in PHP scripting language. The services served the functionalities of the prototype were developed based on the REST style of service communication. The ontology-based contextual annotation service was used to manipulate annotation with contexts and also handle personalised annotations as an adaptation engine.

8.1.3 Evaluation

Evaluation of the CAF implemented as a-PIE prototype was undertaken to prove the benefit of the CAF concept. User evaluation was performed. The result indicated that the users significantly accepted the CAF approach and annotation with contexts concept implemented as a-PIE for evaluation. Although, some users complained that the user interface was not easy to use. The feasibility of this approach was described by assessing the theoretical aspect of the framework against existing related systems.

8.1.4 Summary

This work has presented a new framework of the annotating with context in web pages, called the contextual annotation framework (CAF). The CAF provides adaptable annotations, sharing of annotations and personal repository. These have been achieved by proposed contextual annotation concept, designed ontology using the semantic web languages; RDFS/OWL for the syntactic and semantic interoperability, and separating contextual annotation objects from web content and manipulating by the ontology-based contextual annotation service.

On the whole, the author believes that the research objectives of this work have been achieved. This work has been implemented as a-PIE to conduct the evaluation. However, there have been research issues derived from this work and they require further exploration.

8.2 Research Contributions

This thesis documents several key contributions made to the field of adaptive hypermedia, open hypermedia and the semantic web.

Primarily, this thesis introduces the Contextual Annotation Framework (CAF), a framework that presents the contextual annotation concept in an adaptable web-based system by providing appropriate annotations for users. The concept of contextual annotation means that each annotation object should have contextual objects, or be associated with contextual objects. Context objects keep conditions or criteria for the system to filter contextual annotation objects for an individual user. The annotations to display are determined by settings in the personal profile. The CAF provides the functions of contextual annotation manipulation, personal repository, annotation sharing and reuse of annotations. The framework builds on existing open hypertext, adaptive hypermedia and the Semantic Web principles.

The contextual annotation concept is presented as a way of managing annotations and providing annotations based on context. The contextual annotations are delivered to users according to the users' profiles. This concept can be practically applied to any web-based system with provision of annotation augmentation on the web document and the annotation server. The Ontology-based Contextual Annotation Service is implemented to manipulate the contextual annotation objects. The framework is based on a web services environment.

The formalisation of human-created annotation has been created, expressed as ontological hypertext and implemented them in an open hypermedia and the Semantic Web framework. The contextual annotation objects are manipulated independently as a first-class object and represented in the FOHM structure. The ontology is defined in RDF(S)/OWL.

Finally, the work presents formal evaluation studies which were conducted to confirm whether the concept is beneficial to users and to establish what is the extent and limit of this understanding.

8.3 Future work

Expand the FOHM ontology – Although CAF expanded the FOHM ontology (Gibbins, *et al.*, 2003) to represent annotation structure and use in the framework, the navigational association was designed and implemented only. Since FOHM, described in Section 3.5, and the Semantic Web enable human and machine understanding, the FOHM ontology represented in RDF(S)/OWL is worth designing and exploring. For example, the tour association could be applied to contextual annotation structure in order to allow users to order or reorder contextual annotation objects. In addition, the behaviour object can be attached to a contextual annotation object in order to provide more actions when some behaviours are activated. For example, users can specify different colour highlighting for each annotation type for a particular contextual annotation object.

Augment annotations to the document – in the a-PIE prototype the annotations were displayed as a list of annotations (rendered from FOHM in RDF) for a particular request. There was no implementation of augmenting contextual annotations to the document at a particular annotated region as a link with source (annotated region) and destinations (list of appropriate annotations). For practical use, the implementation of text highlighting and placement of annotations should be considered. The annotations can be displayed in the margin adjacent to highlighted text or displayed in a list of annotation links as a destination of a link. How to represent the highlight in HTML is also worth exploring. In addition, how to support overlap highlighting is also interesting.

Strategies for managing the integrity of annotations (as links) – Some problems might occur when there are annotation manipulations, for example dangling link, orphan link. How to solve these problems is an interesting issue.

Support other media annotation – currently CAF only supports annotation of textual media. However, in hypermedia document there are several kinds of media apart from text, such as video and image. Therefore, annotating and manipulating other media are worth exploring by extending FOHM ontology from CAF ontology. For example, at FOHM Reference, there is LocSpec property storing the location of an annotated region. The LocSpec might be extended by add elapsed time for VDO media or annotated region of image.

Inferencing facts – SHOE agents helped the user find out new facts about their community. Similarly, the inference engine might be integrated into the framework to deliver users new facts. Personalisation algorithms apply known facts about users to customise annotation service. Known user preferences are derived from log files or prior behaviour or from specific statements made in user profiles. In addition, the system may support both adaptable and adaptive annotations users. Adaptive annotations system augments annotations automatically based on user known user preferences. Social network is also an interesting area to provide personalised annotations by learning information from users in the network.

Implement the functionalities as modules for content management system – content management systems (CMS) are popular and easy to develop and install. Developing the module for each function which can be integrated as a module to the CMS should encourage the shareability and reusability.

Future Research Direction –In the short term, continue research will attempt to address the extensions to the completed work listed in the previous section. However, in the longer term, the core research direction adopts an increased focus on the role of Semantic Web technology for data that are linked together to provide more intelligent or adaptable annotations.

References

(Abdullah, 2006) Abdullah, N. A. (2006) An Architecture for Augmenting the SCORM Run-Time Environment with a Personalised Link Service, School of Electronics and Computer Science, University of Southampton, Southampton, UK.

(Ackoff, 1967) Ackoff, R. L. (1967), Management Misinformation Systems. *Management Science*, vol. 14, issue 4, pp. 147 - 156.

(Adler and Van Doren, 1972) Adler, M. J. and Van Doren, C. (1972) *How to Read a Book*: Simon & Schuster Inc; Rev Ed edition.

(Allen, 1990) Allen, R. B. (1990), User models: theory, method, and practice. *International Journal of Man-Machine Studies*, vol. 32, issue 5, pp. 511-543. Available from http://dx.doi.org/10.1016/S0020-7373(05)80032-X.

(Anderson, 1997) Anderson, K., M. (1997) Integrating Open Hypermedia Systems with the World Wide Web, in *Proceedings of the eighth ACM Conference on Hypertext*. Southampton, United Kingdom: ACM Press, 1997.

(Anderson, *et al.*, 1994) Anderson, K. M., Taylor, R. N., and Whitehead, E. J. J. (1994) Chimera: Hypertext for Heterogeneous Software Environments. In Proceedings of the 1994 European Conference on Hypermedia Technology, ECHT'94, Edinburgh Scotland, pp 94-107.

(Anderson, *et al.*, 2000) Anderson, K. M., Taylor, R. N., and Whitehead, E. J. J. (2000), Chimera: Hypermedia for Heterogeneous Software Development Environments. *ACM Transactions on Information Systems*, vol. 18, issue 3, pp. 211-245.

(Ardissono and Goy, 1999) Ardissono, L. and Goy, A. (1999) Tailoring the Interaction With Users in Electronic Shops. In Proceedings of the 7th International Conference on User Modeling (UM99), Banff, Canada, pp. 35-44. Available from http://www.cs.usask.ca/UM99/Proc/ardissono.pdf.

(Aroyo, *et al.*, 2004) Aroyo, L., De Bra, P., and Chepegin, V. (2004) Semantic Web-based Adaptive Hypermedia. In WWW Workshop on Application Design, Development and Implementation Issues in the Semantic Web, New York, USA. Available from http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS//Vol-105/Aroyo-Debra-Chepegin-WWW04-Workshop.pdf.

(Azouaou, *et al.*, 2004) Azouaou, F., Chen, W., and Desmoulins, C. (2004) Semantic Annotation Tools for Learning Material. In Workshop on Applications of Semantic Web Technologies for Adaptive Educational Hypermedia, the Netherlands. Available from http://www.win.tue.nl/SW-EL/2004/AH-SWEL-Camera-ready/SWEL04-AH-PDF/%234-Azouaou-Desmoulins-Weiqin-SWEL-AH.pdf.

(Bailey, *et al.*, 2001) Bailey, C., El-Beltagy, S. R., and Hall, W. (2001) Link Augmentation: A Context-Based Approach to Support Adaptive Hypermedia. In Proceedings of 3rd Workshop on Adaptive Hypertext and Hypermedia, pp 55-62. Available from http://citeseer.ist.psu.edu/bailey01link.html.

(Bailey, *et al.*, 2002) Bailey, C., Hall, W., Millard, D., and Weal, M. (2002) Towards Open Adaptive Hypermedia. In Proceedings of the second International Conference on Adaptive Hypermedia and Adaptive Web-Based System, Malaga, Spain, pp 36-46.

(Bailey, 2002) Bailey, C. P. (2002) An Agent-Based Framework to Support Adaptive Hypermedia, PhD, Electronics and Computer Science, University of Southampton, Southampton. Available from http://eprints.ecs.soton.ac.uk/archive/00007248/.

(Bailey, *et al.*, 2003) Bailey, C. P., Hall, W., Millard, D., and Weal, M. J. (2003), Adaptive Hypermedia through Contextual Structures. *ACM Transactions on Information Systems*. Available from http://eprints.ecs.soton.ac.uk/archive/00009045/.

(Balabanovi and Shoham, 1997) Balabanovi, M. and Shoham, Y. (1997), Fab: Content-based, Collaborative Recommendation. *Commununications of the ACM*, vol. 40, 3, pp. 66-72. Available from http://doi.acm.org/10.1145/245108.245124.

(Baldoni, *et al.*, 2005) Baldoni, M., Baroglio, C., and Henze, N. (2005) Personalization for the Semantic Web. In International Workshop on Personalization on the Semantic Web (REWERSE 2005), Edinburgh, UK, pp 173-212. Available from http://rewerse.net/deliverables/m30/a3-d7.pdf.

(Bechhofer and Goble, 2001) Bechhofer, S. and Goble, C. (2001) Towards Annotation using DAML+OIL. In K-CAP 2001 workshop on Knowledge Markup and Semantic Annotation, Victoria B.C. Available from http://www.cohse.semanticweb.org/papers/index.html

(Bechhofer, *et al.*, 2001) Bechhofer, S., Goble, C., and Horrocks, I. (2001) DAML+OIL is not enough. In SWWS-1, Semantic Web working symposium, Stanford (CA).

(Bechhofer, *et al.*, 2004) Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., and Stein, L. A. (2004) OWL Web Ontology Language Reference. Available from http://www.w3.org/TR/owl-ref/, accessed in 13 May 2007.

(Bellinger, *et al.*, 2004) Bellinger, G., Castro, D., and Mills, A. (2004) The Way of Systems. Available from http://www.systems-thinking.org/index.htm, accessed in 31 March 2007.

(Benjamins, *et al.*, 1998) Benjamins, R. V., Fensel, D., and Perez, A. G. (1998) Knowledge Management through Ontologies. In Proceedings of the 2 nd International Conference on Practical Aspects of Knowledge Management, Basel, Switzerland. Available from http://citeseer.ist.psu.edu/benjamins98knowledge.html.

(Berners-Lee and Cailliau, 1990) Berners-Lee, T. and Cailliau, R. (1990) World Wide Web: Proposal for a HyperText Project. Available from http://www.w3.org/Proposal.html, accessed in 21 March 2007.

(Berners-Lee and Connolly, 1995) Berners-Lee, T. and Connolly, D. (1995) Hypertext Markup Language 2.0 IETF RFC 1866 (Proposed Standard). Available from http://www.ietf.org/rfc/rfc1866.txt, accessed in 11 May 2007.

(Berners-Lee, 1998) Berners-Lee, T. (1998) Semantic Web Roadmap. Available from http://www.w3.org/DesignIssues/Semantic.html, accessed in 30 April 2007.

(Berners-Lee, 2000) Berners-Lee, T. (2000) Semantic Web -- XML2000. Available from http://www.w3.org/2000/Talks/1206-xml2k-tbl/Overview.html

(Berners-Lee, *et al.*, 2001) Berners-Lee, T., Hendler, J., and Lassila, O. (2001), The Semantic Web. *Scientific American*, vol. 279, issue 5, pp. 35-43.

(Berners-Lee, 2003) Berners-Lee, T. (2003) WWW Past & Future. Available from http://www.w3.org/2003/Talks/0922-rsoc-tbl/, accessed in 13 May 2007.

(Berners-Lee, 2005) Berners-Lee, T. (2005) Web for Real People. Available from http://www.w3.org/2005/Talks/0511-keynote-tbl/, accessed in 13 May 2007.

(Berners-Lee, *et al.*, 2005) Berners-Lee, T., Fielding, R., and Masinter, L. (2005) Uniform Resource Identifier (URI): Generic Syntax. Available from http://gbiv.com/protocols/uri/rfc/rfc3986.html, accessed in 11 May 2007.

(Bernstein, 1998) Bernstein, M. (1998) Patterns of Hypertext. In Proceedings of the ninth ACM conference on Hypertext and hypermedia, Pittsburgh, Pennsylvania, United States, pp 21-29. Available from http://portal.acm.org/citation.cfm?id=276630#collab.

(Bieber, 2000) Bieber, M. (2000) *"Hypertext" Encyclopedia of Computer Science (4th Edition)*: Nature Publishing Group.

(Bray, 2007) Bray, D. A. (2007) *Information Pollution, Knowledge Overload, Limited Attention Spans, and Our Responsibilities as IS Professionals*.

(Brickley and Miller, 2007) Brickley, D. and Miller, L. (2007) FOAF Vocabulary Specification 0.9. Available from http://xmlns.com/foaf/spec/, accessed in 12 June 2007.

(Brown, 1987) Brown, P. J. (1987) Turning ideas into products: the Guide system, in *Proceeding of the ACM conference on Hypertext*. Chapel Hill, North Carolina, United States: ACM Press, 1987.

(Bruce, 2005) Bruce, H. (2005), Personal, anticipated information need. *Information Research*, vol. 10, issue 3.

(Brusilovsky, 1994) Brusilovsky, P. (1994) Adaptive Hypermedia: The State of the Art. In HMVR'94 Multimedia, Hypermedia and Virtual Reality, Moscow, pp 24-29.

(Brusilovsky, 1996a) Brusilovsky, P. (1996a), Methods and Techniques of Adaptive Hypermedia. *User Modeling and User Adapted Interaction*, vol. 6, 2-3, pp. 87-129.

(Brusilovsky, 1996b) Brusilovsky, P. (1996b) *Adaptive Hypermedia: an Attempt to Analyze and Generalize*. vol. 1077. Berlin: Springer-Verlag.

(Brusilovsky, 2001) Brusilovsky, P. (2001), Adaptive Hypermedia. *User Modeling and User Adapted Interaction*, vol. 11, pp. 87-110.

(Bush, 1945) Bush, V. (1945), As We May Think. *The Atlantic Monthly*, pp. 101 - 108. Available from http://www.theatlantic.com/doc/194507/bush.

(Campbell and Goodman, 1988) Campbell, B. and Goodman, J. M. (1988), HAM: a General Purpose Hypertext Abstract Machine. *Commun. ACM*, vol. 31, issue 7, pp. 856-861.

(Carlson, 2003) Carlson, C. N. (2003) Information overload, retrieval strategies and Internet user empowerment. . In Proceedings The Good, the Bad and the Irrelevant (COST 269), Helsinki, Finland, pp 169-173. Available from http://eprints.rclis.org/archive/00002248/.

(Carr, *et al.*, 1994) Carr, L., Hall, W., Davis, H. C., De Roure, D., and Hollom, R. (1994), The Microcosm Link Service and its Application to the World Wide Web. Available from http://eprints.ecs.soton.ac.uk/860/02/html/.

(Carr, *et al.*, 1996) Carr, L., De Roure, D., and Hill, G., Ongoing Development of an Open Link Service for the World Wide Web, University of Southampton 1996.

(Carr, *et al.*, 1998) Carr, L., De Roure, D., Hall, W., and Hill, G. (1998), Implementing an Open Link Service for the World-Wide Web.

(Carr, *et al.*, 2001) Carr, L., Bechhofer, S., Goble, C. A., and Hall, W. (2001) Conceptual Linking: Ontology-based Open Hypermedia. In WWW10, 10th World Wide Web Conference, Hong Kong. Available from http://www10.org/cdrom/papers/pdf/p246.pdf.

(Carr, *et al.*, 2002) Carr, L., Kampa, S., De Roure, D., Hall, W., Bechhofer, S., Goble, C. G., and Horan, B. (2002), Ontological Linking: Motivation and Analysis. *CIKM*.

(Carr, *et al.*, 2004) Carr, L., Miles-Board, T., Wills, G., Woukeu, A., and Hall, W. (2004) Towards a Knowledge-Aware Office Environment. In 5th International Conference on Practical Aspects of Knowledge Management. Available from http://eprints.ecs.soton.ac.uk/9847/.

(Carr, 2007) Carr, L., XLink: Open Linking Standard, Course: COMP3016 Hypertext and Web Technologies, University of Southampton, Southampton 2007. Available from https://secure.ecs.soton.ac.uk/notes/comp3016/xml_course/Slides/XLink_files/frame.htm.

(Carro, 2002) Carro, R. M. (2002) Adaptive Hypermedia in Education: New Considerations and Trends. In In Proceedings of the 6th World Multi-conference on Systemics, Cybernetics and Informatics, Orlando, Florida, pp 452-458. Available from http://citeseer.ist.psu.edu/532590.html.

(Chandrasekaran, *et al.*, 1999) Chandrasekaran, B., Josephson, J. R., and Benjamins, V. R. (1999), What Are Ontologies, and Why Do We Need Them? *IEEE Intelligent Systems*, vol. 14, issue 1, pp. 20-26. Available from http://doi.ieeecomputersociety.org/10.1109/5254.747902.

(Chepegin, *et al.*, 2003) Chepegin, V., Aroyo, L., De Bra, P., and Houben, G.-J. (2003) CHIME: Service-oriented Framework for Adaptive Web-based Systems. In In Conferentie Informatiewetenschap, Eindhoven, pp 29-36. Available from http://wwwis.win.tue.nl/infwet03/proceedings/3/.

(Ciancarini, *et al.*, 2002) Ciancarini, P., Folli, F., Rossi, D., and Vitali, F. (2002) XLinkProxy: External Linkbases with XLink, in *Proceedings of the 2002 ACM symposium on Document engineering*. McLean, Virginia, USA: ACM Press, 2002.

(Conklin, 1987a) Conklin, J., A survey of hypertext, Austin 1987a. Available from http://citeseer.ist.psu.edu/conklin95survey.html.

(Conklin, 1987b) Conklin, J. (1987b), Hypertext: An Introduction and survey. *IEEE Computing*, vol. 20, 9, pp. 17-47.

(Conlan, 2004) Conlan, O. (2004) The Multi-Model, Metadata Driven Approach to Personalised eLearning Services, Department of Computer Science, Trinity College, Dublin.

(van Dam, 1988) van Dam, A., (1988), "Hypertext '87: Keynote Address," *Commun. ACM*, vol. 31, 7, pp. 887-895.

(Davies, *et al.*, 2003a) Davies, J., Duke, A., and Sure, Y. (2003a) OntoShare: a knowledge management environment for virtual communities of practice. In Proceedings of the international conference on Knowledge capture Sanibel Island, FL, USA pp 20-27. Available from http://doi.acm.org/10.1145/945645.945652.

(Davies, *et al.*, 2003b) Davies, J., Fensel, D., and van Harmelen, F. (2003b) *Towards the Semantic Web*. London: Wiley.

(Davis, 1995) Davis (1995) Data integrity problems in an Open Hypermedia Link, University of Southampton.

(Davis, *et al.*, 1992) Davis, H. C., Hall, W., Heath, I., Hill, G. J., and Wilkins, R. J. (1992) Towards an Integrated Information Environment with Open Hypermedia Systems. In Proceedings of the ACM Conference on Hypertext (ECHT '92), Milano, Italy, pp 181-190. Available from http://acm.org/pubs/citations/proceedings/hypertext/168466/p181-davis/ (December 1992).

(Davis, *et al.*, 1994) Davis, H. C., Knight, S., and Hall, W. (1994) Light Hypermedia Link Services: A Study of Third Party Application Integration. In Proceedings of the 1994 European Conference on Hypermedia Technology, Edinburgh, pp 44-50.

(Davis, *et al.*, 1999) Davis, H. C., Millard, D. E., Reich, S., Bouvin, N. O., Gronbak, K., Nurnberg, P. J., Sloth, L., Wiil, U. K., and Anderson, K. M. (1999) Interoperability between Hypermedia Systems: The Standardisation Work of the OHSWG. In Hypertext '99, 10th ACM Conference on Hypertext and Hypermedia, Darmstadt, Germany, pp 201-202.

(DCMI, 2006) DCMI (2006) Dublin Core Metadata Initiative: DCMI Metadata Terms. Available from http://dublincore.org/documents/dcmi-terms/, accessed in 6 May 2007.

(De Bra, *et al.*, 1999) De Bra, P., Brusilovsky, P., and Houben, G.-J. (1999), Adaptive Hypermedia: From Systems to Framework. *ACM Computing Surveys*, vol. 31, 4es.

(De Bra, 2000) De Bra, P. (2000), Pros and Cons of Adaptive Hypermedia in Web-Based Education. *Journal on CyberPsychology and Behavior*, vol. 3, 1, pp. 71-77. Available from http://wwwis.win.tue.nl/~debra/cyber.html.

(De Bra, *et al.*, 2004) De Bra, P., Aroyo, L., and Chepegin, V. (2004), The Next Big Thing: Adaptive Web-Based Systems. *Journal of Digital Information*, vol. 5, 1. Available from http://jodi.tamu.edu/Articles/v05/i01/DeBra/#refs

(De Rose, 1989) De Rose, S. J. (1989) Expanding the Notion of Links. In Proceedings of the Hypertext '98 Conference on Hypertext, Pittsburgh, Pennsylvania, USA, pp 249 - 257. Available from http://doi.acm.org/10.1145/74224.74245.

(De Roure, *et al.*, 1999) De Roure, D., El-Beltagy, S., Blackburn, S., and Hall, W. (1999) Integrating Link Resolution Services Using Query Routing. In ACM Hypertext'99 Conference, pp 17-22.

(Decker, *et al.*, 1999) Decker, S., Erdmann, M., Fensel, D., and Studer, R. (1999) Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information. In Proceedings of DS-8, pp 151-169.

(Denoue and Vignollet, 2000) Denoue, L. and Vignollet, L. (2000) An Annotation Tool for Web Browsers and Its Applications to Information Retrieval. In Proceedings of RIAO 2000, Paris, France. Available from http://www.fxpal.com/people/denoue/publications/riao2000.pdf

(DeRose, *et al.*, 2001) DeRose, S., Maler, E., and Orchard, J. (2001) XML Linking. Available from http://www.w3.org/TR/xlink/, accessed in 24 May 2007.

(DublinCore) DublinCore Dublin Core Metadata Initiative. Available from http://dublincore.org/, accessed in 16 April 2007.

(Dzbor, *et al.*, 2004) Dzbor, M., Motta, E., and Domingue, J. (2004) Opening Up Magpie via Semantic Services. In ISWC2004, pp 635-649. Available from http://kmi.open.ac.uk/people/dzbor/public/2004/iswc2004-dzbor-etal.pdf.

(Easterby-Smith and Lyles, 2003) Easterby-Smith, M. and Lyles, M. A. (2003) *The Blackwell Handbook of Organizational Learning and Knowledge Management*: The Blackwell Publishing Ltd.

(El-Beltagy, *et al.*, 2001) El-Beltagy, S. R., Hall, W., De Roure, D., and Carr, L. (2001) Linking in context, in *Proceedings of the twelfth ACM conference on Hypertext and Hypermedia*. Århus, none, Denmark: ACM Press, 2001.

(ELS) ELS Learning Objects & Learning Standards. Available from http://www.learnativity.com/standards.html, accessed in 16 April 2007.

(Engelbart and English, 1968) Engelbart, C. and English, W. K. (1968) A Research Center for Augmenting Human Intellect. In Proceedings of the 1968 Fall Joint Computer Conference, San Francisco, CA, pp 395 - 410.

(Engelbart, 1963) Engelbart, D. C. A conceptual framework for the augmentation of man's Intellect, in *Vistas in information handling, Volume 1* P. D. H. D. C. Weeks, Ed. Washington, D.C.: Spartan Books, 1963, pp. 1-29.

(Espinoza, 1996) Espinoza, F. H., Kristina (1996) An Interactive WWW Interface to an Adaptive Information System. In Proceedings of the Workshop on User Modelling for Information Filtering on the World Wide Web, Fifth International Conference on User Modelling, Hawaii. Available from http://www.sics.se/~kia/espinoza_hook.html.

(Falasconi, *et al.*, 1996) Falasconi, S., Lanzola, G., and Stefanelli, M. (1996) Using Ontologies in Multi-Agent Systems. In the Proceedings of the Tenth Knowledge Acquisition For Knowledge-Based Systems Workshop (KAW'96), Banff, Canada. Available from http://ksi.cpsc.ucalgary.ca/KAW/KAW96/falasconi/index.html.

(Fensel, 2002) Fensel, D. (2002), Ontology-Based Knowledge Management *Computer*, vol. 35, 11, pp. 56-59. Available from http://ieeexplore.ieee.org/search/wrapper.jsp?arnumber=1046975.

(Fountain, *et al.*, 1990) Fountain, A. M., Hall, W., Heath, I., and Davis, H. C. (1990) MICROCOSM: An Open Model for Hypermedia with Dynamic Linking. In the Proceedings of the European Conference on Hypertext. Available from http://citeseer.ist.psu.edu/fountain90microcosm.html.

(Furuta and Urbina, 2002) Furuta, R. and Urbina, E. (2002) On the characteristics of scholarly annotations, in *Proceedings of the thirteenth ACM conference on Hypertext and hypermedia*. College Park, Maryland, USA: ACM Press, 2002.

(Gardner, 2007) Gardner, M. (2007) Using Ontologies to Support the Sharing of Learning Resources. Available from accessed in 16 March 2007.

(Garrett, *et al.*, 1986) Garrett, L. N., Smith, K. E., and Meyrowitz, N. (1986) Intermedia: issues, strategies, and tactics in the design of a hypermedia document system, in *Proceedings of the 1986 ACM conference on Computer-supported cooperative work*. Austin, Texas: ACM Press, 1986.

(Garzotto, *et al.*, 1994) Garzotto, F., Mainetti, L., and Paolini, P. (1994) Adding multimedia collections to the Dexter Model, in *Proceedings of the 1994 ACM European conference on Hypermedia technology*. Edinburgh, Scotland: ACM Press, 1994.

(Gibbins, *et al.*, 2003) Gibbins, N. M., Harris, S. W., Michaelides, D. T., Millard, D. E., and Weal, M. J. (2003) Exploring the Relationship Between FOHM and RDF. In Proceedings of 1st International Workshop on Hypermedia and the Semantic Web, Nottingham, UK. Available from http://eprints.ecs.soton.ac.uk/9388/.

(Gillies and Cailliau, 2000) Gillies, J. and Cailliau, R. (2000) *How the Web Was Born: The Story of the World Wide Web*: Oxford University Press.

(Grønbæk and Trigg, 1999) Grønbæk, K. and Trigg, R. H. (1999) *From Web to workplace: designing open hypermedia systems*: MIT Press.

(Gruber, 1993) Gruber, T. R. (1993) Towards principles for the design of ontologies used for knowledge sharing. In Proceeding of International Workshop on Formal Ontology.

(Guarino, 1998) Guarino, N. (1998) Formal ontology and information systems. In Proceeding of FOIS-98.

(Halasz and Schwartz, 1994) Halasz, F. and Schwartz, M. (1994), The Dexter hypertext reference model. *Communications of the ACM*, vol. 37, 2, pp. 30-39. Available from http://doi.acm.org/10.1145/175235.175237.

(Halasz, 1987) Halasz, F. G. (1987) Reflections on NoteCards: Seven Issues For the Next Generation of Hypermedia Systems, in *Proceeding of the ACM conference on Hypertext*. Chapel Hill, North Carolina, United States: ACM Press, 1987.

(Halasz, *et al.*, 1987) Halasz, F. G., Moran, T. P., and Trigg, R. H. (1987) Notecards in a nutshell, in *Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface*. Toronto, Ontario, Canada: ACM Press, 1987.

(Halasz, 1988) Halasz, F. G. (1988), Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems. *Communications of the ACM*, vol. 31, 7, pp. 836-852.

(Hall, 1994b) Hall, W. (1994b), Ending the Tyranny of the Button. *IEEE MultiMedia*, vol. 1, 1, pp. 60-68.

(Hall, *et al.*, 1995) Hall, W., Carr, L., and De Roure, D. (1995) Linking the World Wide Web and Microcosm. In Proceedings BCS workshop on New Directions in Software Development, Wolverhampton, UK.

(Hall, 2000) Hall, W. (2000), As We May Learn: The Application of Hypermedia in Education. *Computing & Control Engineering Journal*, vol. 11, 4, pp. 173-178.

(Handschuh, *et al.*, 2001) Handschuh, S., Staab, S., and Maedche, A. (2001), CREAM--Creating relational metadata with a component-based, ontology-driven annotation framework. *ACM*, pp. 76-83.

(Handschuh and Staab, 2002) Handschuh, S. and Staab, S. (2002) Authoring and Annotation of Web Page in CREAM. In Proceedings of International World Wide Web Conference, Honolulu, Hawaii, USA. Available from http://citeseer.ist.psu.edu/handschuh02authoring.html.

(Handschuh, *et al.*, 2002) Handschuh, S., Staab, S., and Ciravegna, F. (2002), S-CREAM - Semi-automatic CREAtion of Metadata. *Expert Update*, vol. 5, 3.

(Head, *et al.*, 2000) Head, M., Archer, N., and Yuan, Y. (2000), World Wide Web navigation aid. *International Journal of Human-Computer Studies*, vol. 53, 2, pp. 301-330.

(Heflin, *et al.*, 1999) Heflin, J., Hendler, J., and Luke, S., SHOE: A Knowledge Representation Language for Internet Applications, Department of Computer Science, University of Maryland, Technical Report CS-TR-4078 (UMIACS TR-99-71) 1999.

(Henze, 2005) Henze, N. (2005) E-Learning in the Semantic Web: The Personal Learning Object Readers. In Proceedings of 2nd Joint Workshop on Cognition and Learning through Media-Communication for Advanced e-Learning 2005 (JWSJG05), Tokyo, Japan. Available from http://rewerse.net/publications/description?key=@id&value=%5EREWERSE-RP-2005-109.

(Herman, 2001) Herman, I. (2001) Semantic Web. Available from http://www.w3.org/2001/sw/, accessed in 5 March 2007.

(Hildreth and Kimble, 2002) Hildreth, P. M. and Kimble, C. (2002), The duality of Knowledge. *Information Research*, vol. 8, 1. Available from http://informationr.net/ir/8-1/paper142.html.

(Hornby, 2000) Hornby, A. S. (2000) *Oxford Advanced Learner's Dictionary of Current English*, Sixth edition ed. Oxford: Oxford University Press.

(Hothi, 2000) Hothi, J. (2000) Using an Open Hypermedia System to Develop New Techniques in Adaptive Hypermedia, University of Southampton.

(IEEELOM) IEEELOM IEEE Learning Technology Standards Committee. Available from http://ltsc.ieee.org/wg12/, accessed in 16 April 2007.

(Kahan, *et al.*, 2001) Kahan, J., Koivunen, M.-R., Hommeaux, E. P., and Swick, R. R. (2001) Annotea: An Open RDF Infrastructure for Shared Web Annotations. In Proceedings of the Tenth International World Wide Web Conference, Hong Kong. Available from http://www10.org/cdrom/papers/488/.

(Kampa, 2002) Kampa, S. (2002) Who Are the Experts? E-scholars in the Semantic Web, University of Southampton, Southampton. Available from http://eprints.ecs.soton.ac.uk/7222/.

(Kavcic, 2000) Kavcic, A. (2000) The Role of User Models in Adaptive Hypermedia Systems. In Proceedings of the Electrotechnical Conference, MELECON 2000, 10th Mediterranean, Lemesos, Cyprus, pp 119-122.

(Kerr, *et al.*, 2002) Kerr, A. W., Hall, H. K., and Kozub, S. A. (2002) *Doing Statistics With SPSS* Sage Publications Ltd
(Kirakowski and Corbett, 1993) Kirakowski, J. and Corbett, M. (1993), SUMI: The Software Usability Measurement Inventory. *British Journal of Educational Technology*, vol. 24, pp. 210 - 212.

(Koch and Wirsing, 2002) Koch, N. and Wirsing, M. (2002) The Munich Reference Model for Adaptive Hypermedia Applications. In Proceeding of 2nd International Conference on Adaptive Hypermedia and Adaptive Web-based Systems.

(Koivunen, 2001) Koivunen, M.-R. (2001) Annotea Project. Available from http://www.w3.org/2001/Annotea/, accessed in 12 June 2007.

(Koivunen and Swick, 2001) Koivunen, M.-R. and Swick, R. R. (2001) Metadata Based Annotation Infrastructure offers Flexibility and Extensibility for Collaborative Applications and Beyond. In Proceedings of the First International Conference on Knowledge Capture (KCAP 2001), Victoria, B.C., Canada. Available from http://www.w3.org/2001/Annotea/Papers/KCAP01/annotea.html.

(Library, 2005) Library, N. E. (2005) Communities of Practice. Available from http://www.nelh.nhs.uk/knowledge_management/km2/cop_toolkit.asp, accessed in 8 March 2007.

(Lowe and Hall, 1999) Lowe, D. and Hall, W. (1999) *Hypermedia and the Web: An Engineering Approach*: Wiley & Sons.

(Maedche, *et al.*, 2003) Maedche, A., Motik, B., Stojanovic, L., Studer, R., and Volz, R. (2003), Ontologies for Enterprise Knowledge Management. *Intelligent Information Processing*, pp. 26 - 33. Available from http://doi.ieeecomputersociety.org/10.1109/MIS.2003.1193654

(Maneewatthana, *et al.*, 2005a) Maneewatthana, T., Wills, G., and Hall, W. (2005) Adaptive Personal Information Environment based on the Semantic Web. In Proceedings of Sixteenth ACM Conference on Hypertext and Hypermedia, Salzburg, Austria., pp 214-216.

(Maneewatthana, *et al.*, 2005b) Maneewatthana, T., Wills, G., and Hall, W. (2005) Adaptive Personal Information Environment based on the Semantic Web. In Proceedings of Hypertext 2005 - ACM International Workshop on Adaptive and Personalized Semantic Web, Salzburg, Austria, pp 65-73.

(Marshall, 1998a) Marshall, C. C. (1998) The Future of Annotation in a Digital (Paper) World, in 35th Annual GSLIS Clinic: Successes and Failures of Digital Libraries (Harum and Twidale, eds), 1998.

(Marshall, 1998b) Marshall, C. C. (1998) Toward an Ecology of Hypertext Annotation, in *Proceedings of the ninth ACM conference on Hypertext and hypermedia*. Pittsburgh, Pennsylvania, United States: ACM Press, 1998.

(Marshall, *et al.*, 2001) Marshall, C. C., Price, M. N., Golovchinsky, G., and Schilit, B. N., "Designing E-Books for Legal Research," in *Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries*. Roanoke, Virginia, United States: ACM, 2001.

(Michaelides, *et al.*, 2001) Michaelides, D. T., Millard, D. E., Weal, M. J., and De Roure, D. C. (2001) Auld Leaky: A Contextual Open Hypermedia Link Server. In Proceedings of the seventh Workshop on Open Hypermedia Systems ACM Hypertext Conference, Aarhus, Denmark.

(Miles-Board, 2004) Miles-Board, T. J. (2004) Everything Integrated: A Framework for Associative Writing in the Web, Electronics and Computer Science, University of Southampton, Southampton. Available from http://eprints.ecs.soton.ac.uk/8852/.

(Miles and Brickley, 2005) Miles, A. and Brickley, D. (2005) SKOS Core Guide. Available from http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20050510/, accessed in 13 June 2007.

(Millard, *et al.*, 2000) Millard, D., Moreau, L., Davis, H. C., and Reich, S. (2000) FOHM: A Fundamental Open Hypertext Model for Investigating Interoperability between Hypertext Domains. In Proceedings of the 11th ACM Conference on Hypertext and Hypermedia HT'00, pp 93-102. Available from http://doi.acm.org/10.1145/336296.336334.

(Millard, *et al.*, 2003) Millard, D., Davis, H. C., Weal, M., Aben, K., and De Bra, P. (2003) AHA! meets Auld Linky : Integrating Designed and Freeform Hypertext Systems. In Proceedings of the 14th ACM conference on Hypertext and hypermedia, Nottingham, UK, pp 161-169. Available from http://eprints.ecs.soton.ac.uk/8031/.

(Motta, *et al.*, 2003) Motta, E., Domingue, J., Cabral, L., and Gaspari, M. (2003) *IRS-II: A Framework and Infrastructure for Semantic Web Services*. vol. 2870/2003: Springer-Verlang Heidelberg.

(Nelson, 1965) Nelson, T. H. (1965) Complex information processing: a file structure for the complex, the changing and the indeterminate, in *Proceedings of the 1965 20th national conference*. Cleveland, Ohio, United States: ACM Press, 1965.

(Nelson, 1995) Nelson, T. H. (1995) The heart of connection: hypermedia unified by transclusion, vol. 38: ACM Press, 1995, pp. 31-33.

(Nelson, 1999) Nelson, T. H. (1999), Xanalogical Structure, Needed Now More than Ever:Parallel Documents, Deep Links to Content, Deep Versioning and Deep Re-Use *ACM Computing Surveys* vol. 31, 4. Available from http://xanadu.com/XUarchive/ACMpiece/XuDation-D18.html.

(Nielsen, 1995) Nielsen, J. (1995) *Multimedia and Hypertext: The Internet and Beyond*: Academic Press Professional.

(Noy and McGuinness, 2001) Noy , N. and McGuinness, D. L., Ontology development 101: A guide to creating your first ontology, Stanford Medical Informatics, Stanford, Technical Report KSL-01-05 2001.

(O'Hara, *et al.*, 1998) O'Hara, K., Smith, F., Newman, W., and Sellen, A. (1998) Student Readers' Use of Library Documents: Implications for Library Technologies, in *Proceedings of the SIGCHI conference on Human factors in computing systems*. Los Angeles, California, United States: ACM Press/Addison-Wesley Publishing Co., 1998.

(O'Hara, *et al.*, 2002) O'Hara, K., Alani, H., and Shadbolt, N. R. (2002) Identifying Communities of Practice. In Proceedings of the IFIP 17th World Computer Congress - TC8 Stream on Information Systems: The e-Business Challenge. Available from http://eprints.ecs.soton.ac.uk/6522/01/OHara-IFIP-WCC-2002-camera-ready.pdf.

(Ohene-Djan, *et al.*, 2003) Ohene-Djan, J., Gorle, M., Bailey, C., Wills, G., and Davis, H. (2003) Is It Possible to Devise a Unifying Model of Adaptive Hypermedia and Is One Necessary? In AH2003: Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems, Fourteenth Conference on Hypertext and Hypermedia, Nottingham, UK. Available from http://wwwis.win.tue.nl/ah2003/proceedings/ht-3/.

(Ovsiannikov, *et al.*, 1999) Ovsiannikov, I. A., Arbib, M. A., and McNeill, T. H. (1999) Annotation technology, vol. 50: Academic Press, Inc., 1999, pp. 329-362.

(Paramythis and Stephanidis, 2005) Paramythis, A. and Stephanidis, C. A Generic Adaptation Framework for Web-based Hypermedia Systems, in *Adaptable and Adaptive Hypermedia Systems*, S. Y. Chen and G. D. Magoulas, Eds.: London, UK: IRM Press (an imprint of Idea Group, Inc), 2005, pp. 80-102.

(Pearl, 1991) Pearl, A. (1991) Sun's Link Service: A Protocol for Open Linking. In Proceedings of the Hypertext'89 Conference on Hypertext, Pittsburgh, Pennsylvania, USA, pp 137-146. Available from http://doi.acm.org/10.1145/74224.74236.

(Phelps and Wilensky, 2000) Phelps, T. A. and Wilensky, R. (2000) Robust Intra-document Locations. In Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications netowrking, pp 105-118. Available from http://www9.org/w9cdrom/312/312.html.

(Robson, 2004) Robson, R. (2004) Reusability Guidelines for Authors, Designers and Repositories. Available from http://www.reusablelearning.org/docs/presentations/elearningguild-0404/robson-reusabilityguidelines.ppt, accessed in 9 May 2007.

(Röscheisen and Mogensen, 1995) Röscheisen, M. and Mogensen, C. (1995) Interaction design for shared World-Wide Web annotations, in *Conference companion on Human factors in computing systems*. Denver, Colorado, United States: ACM Press, 1995.

(Röscheisen, *et al.*, 1995) Röscheisen, M., Mogensen, C., and Winograd, T. (1995) Beyond Browsing: Shared Comments, SOAPs, Trails, and On-line Communities. In The Third International World-Wide Web Conference: Technology, Tools and

Applications, Darmstadt, Germany. Available from http://www.igd.fhg.de/archive/1995_www95/papers/88/TR/WWW95.html.

(Schilit, *et al.*, 1998) Schilit, B. N., Gene, G., and Price, M. N. (1998) Beyond paper: supporting active reading with free form digital ink annotations, in *Proceedings of the SIGCHI conference on Human factors in computing systems*. Los Angeles, California, United States: ACM Press/Addison-Wesley Publishing Co., 1998.

(schraefel, *et al.*, 2002) schraefel, m. c., Zhu, Y., Modjeska, D., Wigdor, D., and Zhao, S. (2002) Hunter Gatherer: Interaction support for the creation and management of within-web-page collections. In Proceedings of the 11th international conference on World Wide Web Honolulu, Hawaii, USA pp 172-181.

(schraefel, *et al.*, 2003) schraefel, m. c., Karam, M., and Zhao, S. (2003) mSpace: interaction design for user-determined, adaptable domain exploration in hypermedia. In Proceedings of AH 2003: Workshop on Adaptive Hypermedia and Adaptive Web Based Systems, Nottingham, UK, pp 217-235.

(Seely Brown and Duguid, 1991) Seely Brown , J. and Duguid, P. (1991), Organizational learning and communities-of-practice. *Organisational Science*, vol. 2, 1, pp. 40 - 57.

(Sheth, 1994) Sheth, B. D. (1994) A Learning Approach to Personalized Information Filtering, Massachusetts Institute of Technology. Available from http://agents.media.mit.edu/publications/sheth-thesis.pdf.

(Sidhu, *et al.*, 2005) Sidhu, A. S., Dillon, T. S., Chang, E., and Sidhu, B. S. (2005) Ontology-based Knowledge Representation for Protein Data. In Proceedings of the third IEEE International Conference on Industrial Informatics (INDIN), pp 535 - 539. Available from http://ieeexplore.ieee.org/iel5/10439/33134/01560433.pdf?arnumber=1560433.

(Silva and Oliveira, 2004) Silva, L. and Oliveira, J. P. M. d. (2004) Applying Semantic Web Technologies to Improve Personalization and Achieve Interoperability between Educational Adaptive Hypermedia Systems. In Proceedings 2nd Workshop On Applications Of Semantic Web Technologies For Educational Adaptive Hypermedia In Conjunction With AH'04, Eindhoven, pp 348-353. Available from http://www.inf.ufrgs.br/~palazzo/OAI/04%20SW-EL%20(AH).pdf.

(Smith, *et al.*, 2004) Smith, N., Schmoller, S., and Ferguson, N., Personalisation in Presentation Services, the Joint Information Systems Committee, Technical Report July 2004 2004. Available from http://www.jisc.ac.uk/uploaded_documents/jp-study-15.pdf.

(Staab, *et al.*, 2001) Staab, S., Maedche, A., and Handschuh, S. (2001) An Annotation Framework for the Semantic Web. In Proceedings of The First International Workshop on MultiMedia Annotation, Tokyo, Japan. Available from http://www.aifb.uni-karlsruhe.de/WBS/Publ/2001/CAFfSW_sstetal_2001.pdf.

(Stevens, *et al.*, 2000) Stevens, R., Goble, C. A., and Bechhofer, S. (2000), Ontology-based Knowledge Representation for Bioinformatics. *Briefings in Bioinformatics*, vol.

1, 4, pp. 398 - 414. Available from http://www.cs.man.ac.uk/~stevensr/papers/briefings-ontology.pdf.

(Stojanivic, *et al.*, 2001) Stojanivic, L., Staab, S., and Studer, R. (2001) e-Learning based on the Semantic Web. In Proceedings of the World Conference on the WWW and Internet WebNet2001. Available from http://www.aifb.uni-karlsruhe.de/WBS/Publ/2001/WebNet_lstsstrst_2001.pdf .

(Uren, *et al.*, 2005) Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., and Ciravegna, F. (2005), Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 4, pp. 14 - 28. Available from http://dx.doi.org/10.1016/j.websem.2005.10.002.

(Vasudevan and Palmer, 1999) Vasudevan, V. and Palmer, M. (1999) On Web Annotations: Promises and Pitfalls of Current Web Infrastructure. In Proceedings of the 32nd Hawaii International Conference on System Sciences. Available from http://ieeexplore.ieee.org/iel5/6293/16782/00772663.pdf.

(W3C) W3C HTML Links. Available from http://www.w3.org/TR/html4/struct/links.html, accessed in 16 March 2007.

(W3C, 1992) W3C (1992) Hypertext Terms. Available from http://www.w3.org/Terms.html, accessed in 22 March 2007.

(W3C, 2004a) W3C (2004a) RDF Vocabulary Description Language 1.0: RDF Schema. Available from http://www.w3.org/TR/rdf-schema/, accessed in 29 April 2007.

(W3C, 2004b) W3C (2004b) XML Schema Part 0: Primer Second Edition. Available from http://www.w3.org/TR/xmlschema-0/, accessed in 29 April 2007.

(W3C, 2004c) W3C (2004c) RDF Primer. Available from http://www.w3.org/TR/rdf-primer/, accessed in 29 April 2007.

(W3C, 2006) W3C (2006) Extensible Markup Language (XML) 1.0 (Fourth Edition). Available from http://www.w3.org/TR/REC-xml/, accessed in 29 April 2007.

(W3C, 2007) W3C (2007) XQuery 1.0: An XML Query Language. Available from http://www.w3.org/TR/xquery/, accessed in 29 April 2007.

(Wadge and schraefel, 2001) Wadge, W. and schraefel, m. c. (2001) A Complementary Approach for Adaptive and Adaptable Hypermedia: Intensional Hypertext. In Proceedings of Third Workshop on Adaptive Hypermedia, held in conjunction with the Twelfth ACM Conference on Hypertext and Hypermedia, Aarhus, Denmark, pp 327-334.

(Wang, *et al.*, 2006) Wang, J., Vries, A. P. d., and Reinders, M. J. T. (2006) Unifying User-based and Item-based Collaborative Filtering Approaches by Similarity Fusion. In SIGIR06, Seattle, Washington. Available from http://ict.ewi.tudelft.nl/pub/jun/sigir06_similarityfuson.pdf.

(Weal, 2000) Weal, M. (2000) Visualisation of Hypermedia Systems: An Open Approach, Electronics and Computer Science, University of Southampton, Southampton. Available from http://eprints.ecs.soton.ac.uk/4312/.

(Webster, 2005) Webster, "Merriam-Webster's Dictionary", http://www.m-w.com/,

(Weinreich and Lamersdorf, 2001) Weinreich, H. W. R. and Lamersdorf, W. (2001) Concepts for Improved Visualization of Web Link Attributes. In Proceedings of the Ninth International World Wide Web Conference on Computer Networks, Amsterdam, the Netherlands, pp. Available from http://www9.org/w9cdrom/319/319.html.

(Wenger, 1998) Wenger, E. (1998), Communities of Practice: Learning as a Social System. *The Systems Thinker*, vol. 9, 5. Available from http://www.ewenger.com/pub/pub_systems_thinker_wrd.doc.

(Wenger, *et al.*, 2005) Wenger, E., White, N., Smith, J., and Rowe, K. (2005) *Technology for Communities*: CEFRIO.

(Whitehead, 2000) Whitehead, J. (2000), As we do write: hyper-terms for hypertext. *SIGWEB Newsl.*, vol. 9, 2-3, pp. 8-18.

(Wiil and Whitehead, 1997) Wiil, U. K. and Whitehead, E. J. J. (1997) Interoperability and Open Hypermedia Systems. In Proceedings of the third Workshop on Open Hypermedia Systems, pp 137 - 145. Available from http://www.aue.auc.dk/~kock/OHS-HT97/Papers/wiil.html.

(Wills, 2000) Wills, G. (2000) The Design and Evaluation of Industrial Hypermedia, PhD, School of Electronics and Computer Science, University of Southampton, Southampton. Available from http://users.ecs.soton.ac.uk/gbw/phdthesis/Thesis.html.

(Wills, *et al.*, 2003) Wills, G., Woukeu, A., Carr, L., and Hall, W. (2003) The Need for Deeper Design in a Semantic Web. In Proceedings of HTSW2003 - First International ACM Workshop on Hypermedia and the Semantic Web, at HT03, Nottingham, UK. Available from http://eprints.ecs.soton.ac.uk/8203/.

(Wills, *et al.*, 2004) Wills, G., Fowler, D., Sleeman, D., Crowder, R., Kampa, S., Carr, L., and Knott, D. (2004) Issues in Moving to a Semantic Web for a Large Corporation. In Proceedings of 5th International Conference on Practical Aspects of Knowledge Mangement, Vienna, Austria, pp 378-388. Available from http://eprints.ecs.soton.ac.uk/10179/01/IssuesMovingSemanticWebLargeCorporation.pdf.

(Wolfe, 2000) Wolfe, J., (2000) "Effects of Annotations on Student Readers and Writers.," In Proceedings of ACM Digital Libraries, San Antonio, TX, pp 19 - 26.

(Wu, *et al.*, 1998) Wu, H., Houben, G. J., and De Bra, P. (1998) AHAM: A Reference Model to Support Adaptive Hypermedia Authoring. In Proceedings of the Conference on Information Science, Antwerp, pp 51-76. Available from http://wwwis.win.tue.nl/~debra/infwet98/paper.pdf.

(Yankelovich, *et al.*, 1988) Yankelovich, N., Hann, B. J., Meyrowitz, N. K., and Drucker, S. M. (1988), Intermedia: the concept and the construction of a seamless information environment. *Computer*, vol. 21, 1, pp. 81-96. Available from http://ieeexplore.ieee.org/iel5/2/5791/00222120.pdf?arnumber=222120.

(Yee, 2002) Yee, K.-P. (2002) CritLink: Advanced Hyperlinks Enable Public Annotation on the Web. In Proceedings of Conference on Computer Supported Cooperative Work (CSCW2002). Available from http://zesty.ca/pubs/cscw-2002-crit.pdf.

(Yesilada, *et al.*, 2006) Yesilada, Y., Bechhofer, S., and Horan, B. (2006) Personalised Dynamic Links on the Web. In First International Workshop on Semantic Media Adaptation and Personalization (SMAP'06), pp 7-12. Available from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4041951.

(Yoo and Bieber, 2000) Yoo, J. and Bieber, M. (2000) Towards a Relationship Navigation Analysis. In Proceedings of the thirty-third Hawaii International Conference on System Sciences, pp 6066. Available from http://doi.ieeecomputersociety.org/10.1109/HICSS.2000.926887.

Appendix A User’s Evaluation

User’s Evaluation

In the experiment there are two systems:

System 1: web-based annotation without context. The system provides adding annotations and displaying all annotations for a particular URL or all annotations available without any context support. Users cannot change the setting.

System 2: web-based annotation with contexts, and personal repository. The system provides annotations creation with contexts support. Users can select how they want annotations to be presented with the predefined (two dimensions) preferences: topic and expertise level. There are four topics and three expertise levels. The topics are about the web development. Expertise levels are beginner, intermediate, and advance. The system also provides personal repository for user to keep interested annotations with personal contexts. This prototype might help users to control annotations and provides personal information environment.

The objective of the task one in this experiment is to compare each system against each other. In this case **the time to finish tasks** is used as a factor. Therefore do not worry if you cannot finish the questions. They are there to see if you can find the answers for them, we are NOT testing your understanding of the subject domain in any way. While task two and task three are features provides in the system 2 (a-PIE) only.

PROTOCOL FORM

Background, research objectives

Background

Knowledge is an essential asset for any organisations (Benjamins, et al., 1998). Knowledge can be tacit (knowledge in human mind) or explicit, documented or procedural, for example, or it can be information, experiences or skills which can be managed through technology (Hildreth and Kimble, 2002). According to Nonaka, explicit knowledge (or information) is knowledge that is easily expressed, captured, stored and reused. Al-Hawamdeh mentioned that knowledge in books or journals will be useful and usable when it is read, manipulated and communicated from one person to another (Al-Hawamdeh, 2002). However, "knowledge acquired through experience cannot get reused because it is not shared in a formal way" (Angus, et al., 1998). Knowledge management is a process of knowledge gathering by adding value to previous knowledge (Duffy, 1999), knowledge organisation and structuring and knowledge maintenance.

According to Ovsiannikov users create annotations to remember the main aspects of a document which reminds easily with access through the annotation (Ovsiannikov, *et al.*, 1999). Critical remarks, questions, notes and ideas reflect personal opinion and support readers in thinking about the document content. Devising content in personal words helps in clarifying certain aspects of a paper, and the sharing of annotations is important in a collaborative environment. Annotations support understanding, memorisation and later retrieval (Schilit, *et al.*, 1998) (Vasudevan and Palmer, 1999). Annotation is an important part of *active reading* (Adler and Van Doren, 1972).

A scenario will help explain the motivation behind the research. Suppose a user in a virtual community is looking for information from a community or organisation's web site. They find an interesting piece of information and would like to keep it for reference later in a personal repository. On occasions, they would like to add annotation to particular information snippets and would like to record the context before storing it in their repository. They would also like to share this annotation with others in their community web site. As this is a diverse people in a community they annotate the information with the level of expertise (e.g. expert intermediate or beginner) the reader requires in order to understand the information. Then depending on the reader's profile (e.g. knowledge background, expertise) only those authorised by the original user will see the annotations annotated by others.

Annotation Definition

The definition for the term **annotation** is that given by the W3C in that an annotation is any object; such as notes, comments or ideas, that is attached to a web document or selected part of the web document. Useful annotations can be invaluable knowledge for other people in a community. Therefore it is beneficial if annotations can be shared with others. Generally hyperlinks in web documents are hard-coded and embedded in the content of the document. Therefore links are difficult to author and there is a lack of ownership. In order to provide shareability and reusability of annotations, this research attempted to permit the shareability and reusability of annotations by using open hypertext technology and utilising the Semantic Web technology to better support the annotation.

Information overload problem occurs when users have got too much information to absorb. Most of annotation systems have no adaptable functions; therefore all annotations will be given to users. Yet different people have different preferences or interests. Therefore, the information overload problem should be taken to be considered in annotation systems in order to provide relevant annotations to users.

Research objectives

The objective of this experiment is to compare two similar systems that can be used for annotating the web. The proposed solution of the research is to consider "**contextual annotation and features of the prototype**" instead of user interface of the web-based annotation system. Research objectives are:

Understand the context role of annotation for the online community web site.

Whether it is beneficial to combine open hypermedia, adaptive hypermedia and the Semantic Web technology to provide adaptable, shareable and reusable annotations for users on the online community web site.

The issues involved are

Does the user desire to set context to annotations (e.g. for categorisation purpose)?

Is the use of context useful for making annotation on the web?

Are adaptable annotations, shared annotations, and personal repository of annotation useful?

Recruitment

The participants of these experiments are voluntary postgraduate students or researchers at the University of Southampton. The participants have background in information technology and non-information technology related background. The research targets around 25-30 users. The users divided into two groups. The first group perform the experiment with the web-annotation without contexts, and follow by the web-annotation with contexts. The second group perform the experiment with the web-annotation with contexts first, and follow by the web-annotation without contexts.

The experiment takes about 30 minutes. The participants seat at the desk at all times during the tasks. Water is provided to make sure that the environment is comfortable for them. The participants have to fill in the pre-evaluation questionnaire. After the experiments, the users fill in the questionnaire.

Risks and benefits

There are no physical risks to the participants. Use of the system will be optional. User can terminate the experiment at any time. The benefits of an adaptable personal information environment is to help users: annotate their thought while they are reading hypermedia on an online-community web site, share their ideas with others and keep the particular ideas on the community web site to personal repository.

Privacy and confidentiality

All information that you provide on this questionnaire will be kept strictly confidentially. The only information that will be shared is the findings and discussions according to the questionnaire. This questionnaire will be handled anonymously as all the information you provide will be mainly used as a part of a PhD research on whether it is beneficial and feasible to provide contextual annotations, shareability and reusability of annotations for the online community web site.

Please sign this document to give the research permission to perform the experiments and use findings for research purposes. If you have any queries please do not hesitate to contact Thanyalak Maneewatthana at the Learning Societies Lab, Building 32 (EEE), level 3, room 3069, or via email: tm03r@ecs.soton.ac.uk.

Thank you in advance for your support.

Name:______________________________________ Date: ____________________

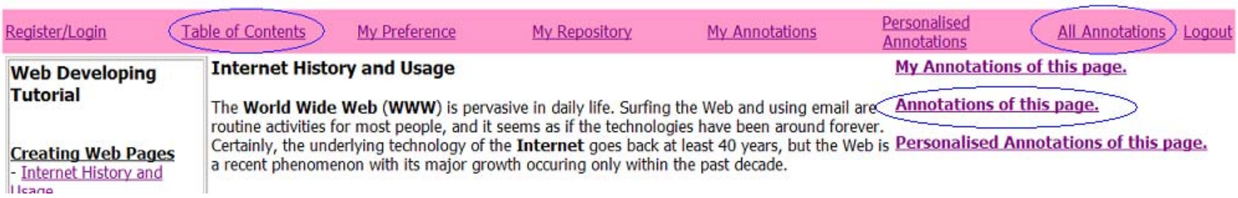
Witness: ____________________________________

System 1 (annotation with no preferences)

<u>Task 1</u>

This task is to investigate the effectiveness, efficiency and usability of the contextual annotation concept.

Please use '**<u>Annotations of this page</u>**' (from the menu on the right of each page) or '**<u>All Annotations'</u>** (from the menu on the top).



Noted that all answers were **already on the list of annotations**. Please **do not use 'control+F'** key to find the exact word matched.

Register and log in to the prototype system. (username:______)

Explore the web site generally for 2 minutes.

Use the web site to answer the following questions, or as many as possible. Please do not forget to write down the time when you start and are done.

Time start: ___

Question 1: Who did coin the word "hypertext"? In which year? ___________

Question 2: what is triadic colour? ___________________________________

Question 3: Which graphic format is the most widely supported? ___________

Question 4: Which function delivers the results of processing to the user in a correct, timely, and appropriately formatted fashion? ___________________________

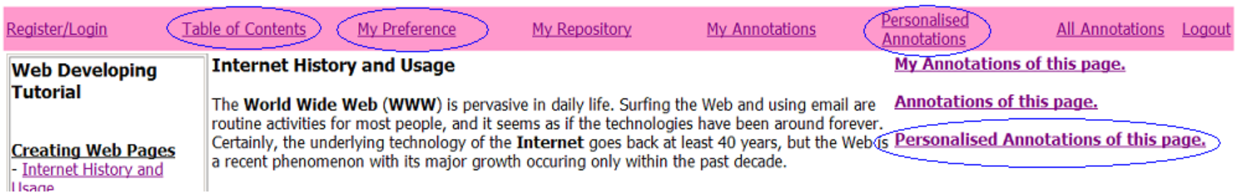
Question 5: Which figure shows "rasterizing outline fonts"? _______________

Time finish: ___

System 2 (annotation with preferences, and repository : a-PIE)

Task 1: This task is to investigate the effectiveness, efficiency and usability of the contextual annotation concept. You are allowed to **set 'My Preference'** to find annotations based on your preferences.

You can find information from menu '**Personalised Annotations'** or '**Personalised Annotations**' from menu on the right of each page.



Noted that all answers were **already annotated**. Please **do not use 'control+F'** key to find the exact word matched.

Register and log in to the prototype system. (username:______)

Explore the web site generally (about 2 minutes).

Use the web site to answer the following questions.

Time start: ______

Question 1: When (in which year) did the Internet begin? ________________________

Question 2: What is "split-complementary colour"? ____________________________

Question 3: Which graphic format improves image quality and offers better image compression? ____

Question 4: Which function permits users to interact with the system? ______________

Question 5: Which figure shows "typical core fonts installed on desktop computers"?

____

Time finish: ___

Task 2: Select (click) any topic from the menu on the left. Imagine that while you are browsing the web site, you found interesting texts and want to annotate it for reference later.

You can create an annotation by:

Copy texts you want to annotate (like when you highlight text)

At the bottom of that page, there is a box for making an annotation

Paste the selected text to the field 'annotated text', select relation type, fill in title, and annotation, set context as you want. As you want to annotate for yourself only, then select authorised group as 'private'. For example, you might note summary or comments for the selected text.

Press 'submit annotation'.

You can see your annotations from 'My Annotations of this page' at the menu on the right

Any Comments about this function?___________________________

Is it useful or helpful?____

Task 3 : Imagine that when you read the web site, you found an interesting annotation created by other and you want to keep it in your repository for reference later. Therefore, you want to add that annotation to your repository.

To do:

Table of Contents: Select 'Web System Models'.

At menu on the right, select the 'Annotations of this page'. You will see all annotations created by others.

At the first annotation, 'Information Delivery Model' created by iampiglet, you think it is interesting comment so you want to keep it to your repository. Select an annotation by clicking on '<u>Add to My Repository'</u> button.

Set <u>context</u> you want. You think it is about 'Web Developing: Model/ Architecture: Intermediate'. So you select this context. And then click on '<u>submit annotation'</u> button.

The annotation will be stored in your repository with the context you select.

Any Comments about this function? ________________________________

Is it useful or helpful?____

Thank you for your participation. Please fill in the questionnaire.

Questionnaire:

The questionnaire filled by subjects during the session asking feedback on the annotations with preferences, my repository and my annotations (a-PIE). All information will be kept confidentially.

Thank you very much for going through the experiment. Please enter your responses through the questionnaire.

The proposed solution of the research is to consider "<u>contextual annotation framework</u>" instead of the user interface of the web-based annotation system.

Please answer all the questions (✔).

Personal profiles

1.	Background: Gender: Female [] Male []
2.	Current level of education: ____________ Please specify area (✔): Computer Science/IT or IS related Management Other science or engineering Others : please specify____________________________________
3.	How often do you use a computer (computer skill) Once a week Every day a little bit Every day a lot
4.	How frequency do you surf the internet: None Once a week Every day a little bit Every day a lot
5.	How often do you use online community web site: e.g. web forum (web board), chat, blog, etc.? None Once a week Every day a little bit Every day a lot Please specify the purpose of use: (e.g. communication, share, exchange idea, relax, get more information, etc.)
6.	Do you annotate (e.g. highlight) while you are reading papers (e.g. books, papers, reports, etc.)? None Little Some Always Please specify the purpose of making annotations: ____________________________________
7.	How much experience did you have using web annotation (e.g. bookmark)? None Little (once a week or less) Some (every day a little bit) Much (every day a lot) If yes, please specify the web annotation you ever used:_________________
8.	Do you feel comfortable with sharing opinions with friends in an online community? Not at all Not very Quite Very

Please select (✔) one of the following scales to reflect upon how you feel about the beneficial of the adaptable annotation framework (CAF) as prototyped as adaptable personal information environment (a-PIE): making annotations with preferences, my annotations and my repository.

Affect – user's emotions toward the usage of the system					
1. I enjoyed interacting with the system: annotation with preferences (a-PIE).	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
2. The system (a-PIE) was confusing to use.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
3. When I found an interesting annotation created by others, I thought it was useful that I could add other's annotation to my repository.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
4. With the functionalities of the annotation system (making annotations with preferences, my repository, my annotations) is the one that I would want to use on a regular basis.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

Control – the degree to which the user feels that they are in control.					
5. In order to manipulate annotations (e.g. add, search annotations) I liked that there were some predefined preferences in my preference to choose.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
6. Apart from predefined preferences, I want to create my own preferences (e.g. concept, event, thing, etc.).	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
7. I liked that I could set personal preferences in order to get personalised annotations.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
8. When I changed the preferences, the annotations reflected what I set.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
9. I would like a greater level of control of annotations (i.e. how the annotation was structured, number of annotations per page).	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

Efficiency – the degree to which users can complete tasks in a direct and timely fashion.					
10. To obtain the annotations I needed from the system, the annotations with preferences was easier than the annotations with no preferences.	Strongly Agree	Agree	Neutral	Disagree	StronglyDisagree
11. To get annotation I needed from the system (annotations with preferences) was straightforward.	Strongly Agree	Agree	Neutral	Disagree	StronglyDisagree
12. The personalised annotations reflect to what I set in my preferences.	Strongly Agree	Agree	Neutral	Disagree	StronglyDisagree
13. To get annotation I needed in the system (annotations with preferences) was awkward.	Strongly Agree	Agree	Neutral	Disagree	StronglyDisagree

Helpfulness – the extent to which the system assists the user in a situation.					
14. The 'my repository' for storing interesting annotations was helpful.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
15. The 'my annotations' for storing all annotations created by myself was useful.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
16. To classify annotations, it was useful to allow user to add contexts to annotations.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
17. I could understand how to use the system (annotation with preferences) and found this system helpful.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
18. The contexts set in 'my preference' helped me find the annotations based on my preferences.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
19. The contexts in 'my preferences' were useful as it allowed me to select the annotations to be displayed.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
Learnability – the degree to which the system is easy for users to learn how to use.					
20. Learning to use the system (annotation with contexts: a-PIE) was straightforward.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
21. I found the system (annotation with preferences) easy to learn and use.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

22. The user interface was not very easy to use.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
Navigation – the ability that users can move around the system.					
23. Specifying the personal preferences could help me to navigate the annotations and information space.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
Comprehension – the degree to which users can understand the interaction with the system.					
24. I understood the interaction with the system.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
25. The system was clearly presented and understandable.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
26. I understood the concept of the system (annotation with preferences, my annotations, my repository).	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
27. The information displayed was consistent.	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

Other comments:

28. What is your opinion about the contexts in the system? Is it helpful to have more than dimensions of contexts in my preferences, for instance, in the 'web developing' there was a more specific concept like 'model/architecture' for user who has 'intermediate' of knowledge background, or have another dimension like 'language' for a particular concept?
29. Are there any additional features you would like to see in the web annotation system? Yes No Comments:
30. Overall reactions to the features of the system: making annotations for the web site, set preferences for annotations and see preferences based on preferences, my repository, my annotations (not user interface). Difficult Neutral Satisfactory
31. Please use the space below to add any additional comments.

Thank you very much for your time and kind co-operation.

All information given will be kept confidentially.

Appendix B Data from Experimental Study

Raw Data

User	28.What is your opinion about the contexts in the system? Is it helpful to have more than dimensions of contexts in my preferences, for instance, in the 'web developing' there was a more specific concept like 'model/architecture' for user who has 'intermediate' of knowledge background, or have another dimension like 'language' for a particular concept?	29.Are there any additional features you would like to see in the web annotation system?/comments?
1	There should be a standard for contexts	Create my own contexts
2		It is an interesting idea and it would be useful to access information on the web sites that allow users to mark/take note on the content and refer back later.
3	it requires expert to do.	better user interface
4		user can create new preferences
5	context adding that will be enough to set justify word such as keyword, people that will be enough. (exp level might not need)	the presentation of my repository should separate according to preference or topic in a way that easy to read.
6		
7		delete annotations, colourful of annotation to identify important words or phrase
8	yes	colour highlight
9	yes	
10	in the current articles, they are enough. Maybe more in a difficult circumstance.	may be add the flexibility so that users can create context
11	neutral	highlight the screen when chose
12	yes, may be it's good to have more pferences with more categories inseparate sections and at the same time user can click as many as they like.	
13	It s not necessary to identify the person's knowledge level about the contents. It'll be more useful if the user can type or read other users' annotations in his/her language.	

14	I don’t think it's useful. It will make 'my preferences function' more complicated. Who are going to define which topic is for beginners or intermediate users (readers), etc.	be able to define 'keywords/contexts' in my preference, be able to select personalised annotation list of whom I want to read.
15	the system (knowledge backgroun) is interesting, but language might not be useful.	
16	sort of letter or group	word searcher
17	Yes, contexts with many dimensions concept is useful, but difficult for user to handle it.	automatic annotation
18	agree	image
19		
20	Knowledge background is useful. It similars to teach students step by step.	Annotation image/figure
21	yes, this would make the searching process more effectively and faster	the additional of user's reputation system
22	it is good for every level of usrs since they can understand the information easier and faster	in my repository, it 'll be helpful if I can have a "library or directiory" to put my annotations into them
23	it's useful to sub-categorise the group. However it's not necessary to do because it is difficult to separate the group of knowledge, background	keyword searching
24	for my preferences, it's useful and save time because we can set the context specifically. On the other hand, it may become difficult to find the text if we break it into too specific context.	perhaps we can set the link website that also relates to the text we are interested
25	It is useful because it helps user classify topics, and it makes user easy to find information again.	FAQ
26	it's helpful but if we want to find it quickly, the same name 'web developing' in many places can make us confuse.	highlight on the original paper with different colour for a different context

User	**31.Please use the space below to add any additional comments.**	**System:comment1**	**System: comment2**
1			
2	Annotations with colours		
3			
4	Comparing between annotations with contexts and with no contexts, the first one was faster.		
5	the concept is fine.	it's ok, but need to have compact interface	
6			
7	the annotation framework might be useful for users because they can add important words and it will be useful to remind people to remember words or phrase	I think that'll help and remind users	it's useful for others people who also are interested in the same annotations

8			
9		better user interface, no copy and paste in order to annotate.	
10			
11			good function
12	I like the personal note option. However, I think it's good to have any kind of sorting in that area.	function should provide a highlight after user selected the annotated text, to give a better result in terms of user interface	
13			
14			
15		this function is great. I use more shorter time to answer the questions.	this function is excellent. Readers can share their ideas each other.
16			
17		It is useful for summary and presentations in case the annotations are good	it's quite good.
18	good	Refer back, easy to find	
19		Allow user to create own contexts, and predefined from the system	very useful
20	willing to download and install on personal machine.		
21			
22		practical and easy to use	help a lot for finding our interesting subjects
23	It is difficult to classify knowledge background. Grouping of annotations by contexts makes it easy to find. However, if there are too many contexts, it might be too complicated. Create contexts by ourselves might be difficult to find it later.	keyword searching program would help the readers easily access to wanted context or data	this can help collect other's view which is very useful.
24	for intermediate of knowledge background, it should be great if we can set more what we want to find. However, it's probably complicated to understand how to use this function.	definitely useful, we can save time for reading in second times	useful and reasonable that we can set context by ourselves even they are created by others because

			we are may interested in those texts in different points from others
25	This kind of annotations for the web would be useful for e-learning. It reduces time to read this web again by reading the annotations and also gets relevant information. However, the user interface should not be too complicated.	It is useful for refer back.	Adding other annotations into personal repository is useful in that it reduces time if that annotaitons are good and fit our needs.
26	1. after select the context we want, we have to click to bottom below to confirm firm and then click another place to see a page with annotations if the process of clicking the bottom below is deleted it'll be better. 2. the page for selecting annotation, if the name of the context can be put in one column it may be better for comparing and finding the name, 3. after choosing the cotext and the annotation we want if the list of the annotation can be put together with the original paper and having highlight in the context in colours may be better.	useful: it's good when we want to come back and reread it again so that we can see what we think that it's important in the past and compare with our opinion now.	useful: it's help for comparing people's opinion (which others think that important, if it s the same we can spend less time to keep the important fcontext for ourselves)

	User	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
	Affect (1:strongly disagree, 2: disagree, 3:neutral, 4:agree, 5:strongly agree)																										
1	I enjoyed interacting with the system: annotation with preferences (a-PIE).	5	4	2	5	5	5	4	4	5	4	3	4	4	3	5	4	4	4	3	4	4	5	4	4	5	4
2	The system (a-PIE) was confusing to use.	4	4	4	4	3	3	4	3	4	4	4	4	4	4	4	2	3	4	2	3	4	5	3	4	4	4
3	When I found an interesting annotation created by others, I thought it was useful that I could add other's annotation to my repository.	5	5	4	4	5	5	5	3	4	4	4	4	5	4	5	5	4	4	4	5	4	5	5	5	4	5
4	With the functionalities of the annotation system (making annotations with preferences, my repository, my annotations) is the one that I would want to use on a regular basis.	5	5	4	5	5	5	4	4	4	5	4	5	4	4	5	4	4	4	3	5	3	5	5	4	4	3
	Control																										
5	In order to manipulate annotations (e.g. add, search annotations) I liked that there were some predefined preferences in my preference to choose.	4	5	2	4	4	5	4	4	4	3	4	4	4	3	5	4	4	4	4	5	3	5	3	4	4	5
6	Apart from predefined preferences, I want to create my	5	5	4	5	4	5	4	4	4	4	4	5	4	5	5	4	4	4	4	4	4	4	5	5	4	5

	own preferences (e.g. concept, event, thing, etc.).																										
7	I liked that I could set personal preferences in order to get personalised annotations.	5	5	4	5	5	5	4	4	5	4	4	5	4	5	5	4	3	4	4	5	4	5	5	5	4	4
8	When I changed the preferences, the annotations reflected what I set.	4	5	4	5	5	4	4	4	5	4	4	5	4	5	5	3	4	3	4	5	4	5	3	5	4	5
9	I would like a greater level of control of annotations (i.e. how the annotation was structured, number of annotations per page).	5	4	4	4	3	4	4	5	4	4	4	5	4	2	3	3	4	3	4	4	4	4	3	4	3	5
	Efficiency																										
10	To obtain the annotations I needed from the system, the annotations with preferences was easier than the annotations with no preferences.	5	4	4	5	5	5	5	4	4	4	4	5	4	4	5	4	4	5	4	4	4	5	5	5	5	5
11	To get annotation I needed from the system (annotations with preferences) was straightforward.	5	5	4	5	5	3	4	4	4	4	4	5	4	4	5	4	4	4	3	4	4	5	4	4	4	4
12	The personalised annotations reflect to what I set in my preferences.	5	5	4	5	5	3	4	5	4	4	4	5	4	5	5	4	4	4	4	4	4	5	4	4	4	5
13	To get annotation I needed in the system (annotations with preferences) was not awkward.	4	4	4	4	5	3	3	3	4	4	3	3	4	3	4	2	2	3	3	4	2	5	3	4	3	3
	Helpfulness																										
14	The ‘my repository’ for storing interesting annotations was	5	5	4	5	4	4	5	4	4	4	4	4	4	4	5	3	4	4	4	5	4	5	5	4	4	4

	helpful.																										
15	The ‘my annotations’ for storing all annotations created by myself was useful.	5	5	4	5	5	4	4	5	4	4	4	4	4	5	5	3	4	4	4	4	4	5	4	5	4	5
16	To classify annotations, it was useful to allow user to add contexts to annotations.	5	5	4	5	5	4	4	4	4	5	3	4	4	5	5	4	4	5	4	5	4	5	4	4	4	5
17	I could understand how to use the system (annotation with preferences) and found this system helpful.	4	5	4	5	5	4	4	5	4	4	4	4	4	4	4	4	4	4	4	4	4	5	4	5	5	5
18	The contexts set in ‘my preference’ helped me find the annotations based on my preferences.	5	5	4	5	5	4	4	5	4	4	4	5	4	4	5	4	4	5	4	4	5	5	5	4	5	5
	Learnability																										
20	Learning to use the system (annotation with context: a-PIE) was straightforward.	4	4	4	4	4	4	4	4	4	4	3	4	4	4	4	3	4	4	3	4	4	5	4	4	4	4
21	I found the system (annotation with preferences) easy to learn and use.	4	5	4	5	4	4	4	4	4	4	4	4	4	3	5	3	3	4	3	4	4	5	4	5	4	4
22	The user interface was very easy to use.	4	4	4	4	4	4	4	4	4	4	3	3	4	1	4	3	2	4	3	4	4	5	2	4	4	4
	navigation																										
23	Specifying the personal preferences could help me to navigate the annotations and information space.	4	5	4	5	5	4	4	3	4	4	4	4	4	4	5	4	4	4	4	4	4	5	5	4	5	4
	Comprehension																										

24	I understood the interaction with the system.	5	5	4	5	5	4	4	4	4	4	4	4	4	3	4	4	3	4	3	4	4	5	5	4	4	5
25	The system was clearly presented and understandable.	4	4	4	5	5	4	4	4	4	4	3	5	4	3	4	4	3	5	3	4	4	5	4	4	4	5
26	I understood the concept of the system (annotation with preferences, my annotations, my repository).	5	5	4	5	5	4	4	4	4	4	4	5	4	4	5	4	4	4	3	4	4	5	4	5	4	5
27	The information displayed was consistent.	4	5	4	5	5	4	4	3	4	4	3	5	4	4	5	4	4	4	4	4	4	5	5	4	4	5
30	Overall reactions to the features of the system: making annotations for the web site, set preferences for annotations and see preferences based on preferences, my repository, my annotations (1:difficult, 2: neutral, 3:satisfactory)	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	2	3	3	2	3	3	3	2	3	3	3

Appendix C Ontology

CAF ontology


```
<?xml version="1.0"?>
<rdf:RDF xml:base="file:/D:/AppServ/www/caf/ontology/caf.owl"
xmlns:caf="http://localhost/caf/ontology/caf.owl#"
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:fohm="http://localhost/caf/ontology/fohm.owl#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:skos="http://www.w3.org/2004/02/skos/core#">
  <owl:Ontology rdf:about="file:/D:/AppServ/www/caf/ontology/caf.owl"/>
  <owl:Class rdf:about="http://localhost/caf/ontology/caf.owl#Group"/>
  <owl:Class rdf:about="http://localhost/caf/ontology/caf.owl#Repository"/>
  <owl:Class rdf:about="http://localhost/caf/ontology/caf.owl#User">
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="http://xmlns.com/foaf/0.1/member"/>
        </owl:onProperty>
        <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
        <owl:onProperty>
          <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/caf.owl#interest"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
        <owl:onProperty>
          <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/caf.owl#hasRepository"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
```

```
            <owl:onProperty>
               <owl:ObjectProperty rdf:about="http://xmlns.com/foaf/0.1/nick"/>
            </owl:onProperty>
            <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
         </owl:Restriction>
      </rdfs:subClassOf>
   </owl:Class>
   <owl:Class rdf:about="http://localhost/caf/ontology/fohm.owl#Data">
      <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
      <rdfs:subClassOf>
         <owl:Restriction>
            <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
            <owl:onProperty>
               <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/fohm.owl#hasContext"/>
            </owl:onProperty>
         </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
         <owl:Restriction>
            <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
            <owl:onProperty>
               <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/caf.owl#authorisedGroup"/>
            </owl:onProperty>
         </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
         <owl:Restriction>
            <owl:onProperty>
               <owl:ObjectProperty rdf:about="http://purl.org/dc/elements/1.1/creator"/>
            </owl:onProperty>
            <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
         </owl:Restriction>
      </rdfs:subClassOf>
   </owl:Class>
   <owl:Class rdf:about="http://www.w3.org/2004/02/skos/core#Concept"/>
   <owl:Class rdf:about="http://localhost/caf/ontology/caf.owl#ExpertiseLevel"/>
   <owl:Class rdf:about="http://localhost/caf/ontology/caf.owl#Context">
      <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
      <rdfs:subClassOf>
         <owl:Restriction>
            <owl:onProperty>
               <owl:ObjectProperty rdf:about="http://purl.org/dc/elements/1.1/subject"/>
            </owl:onProperty>
            <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
```

```
          </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
          <owl:Restriction>
            <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
            <owl:onProperty>
              <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/caf.owl#hasExpertiseLevel"/>
            </owl:onProperty>
          </owl:Restriction>
        </rdfs:subClassOf>
      </owl:Class>
      <owl:ObjectProperty rdf:about="http://purl.org/dc/elements/1.1/creator">
        <rdfs:domain rdf:resource="http://localhost/caf/ontology/fohm.owl#Data"/>
        <rdfs:range rdf:resource="http://localhost/caf/ontology/caf.owl#User"/>
      </owl:ObjectProperty>
      <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/caf.owl#hasPersonalContext">
        <rdfs:domain rdf:resource="http://localhost/caf/ontology/caf.owl#Repository"/>
        <rdfs:range rdf:resource="http://localhost/caf/ontology/caf.owl#Context"/>
      </owl:ObjectProperty>
      <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/fohm.owl#hasContext">
        <rdfs:range rdf:resource="http://localhost/caf/ontology/caf.owl#Context"/>
        <rdfs:domain rdf:resource="http://localhost/caf/ontology/fohm.owl#Data"/>
      </owl:ObjectProperty>
      <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/caf.owl#hasRepository">
        <rdfs:range rdf:resource="http://localhost/caf/ontology/caf.owl#Repository"/>
        <rdfs:domain rdf:resource="http://localhost/caf/ontology/caf.owl#User"/>
      </owl:ObjectProperty>
      <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/caf.owl#authorisedGroup">
        <rdfs:range rdf:resource="http://localhost/caf/ontology/caf.owl#Group"/>
        <rdfs:domain rdf:resource="http://localhost/caf/ontology/fohm.owl#Data"/>
      </owl:ObjectProperty>
      <owl:ObjectProperty rdf:about="http://localhost/caf/ontology/caf.owl#hasData">
        <rdfs:domain rdf:resource="http://localhost/caf/ontology/caf.owl#Repository"/>
        <rdfs:range rdf:resource="http://localhost/caf/ontology/fohm.owl#Data"/>
      </owl:ObjectProperty>
      <owl:ObjectProperty rdf:about="http://xmlns.com/foaf/0.1/member">
        <rdfs:domain rdf:resource="http://localhost/caf/ontology/caf.owl#User"/>
        <rdfs:range rdf:resource="http://localhost/caf/ontology/caf.owl#Group"/>
      </owl:ObjectProperty>
      <owl:ObjectProperty rdf:about="http://purl.org/dc/elements/1.1/subject">
        <rdfs:domain rdf:resource="http://localhost/caf/ontology/caf.owl#Context"/>
        <rdfs:range rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
      </owl:ObjectProperty>
      <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/caf.owl#hasExpertiseLevel">
```

```
      <rdfs:range rdf:resource="http://localhost/caf/ontology/caf.owl#ExpertiseLevel"/>
      <rdfs:domain rdf:resource="http://localhost/caf/ontology/caf.owl#Context"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:about="http://localhost/caf/ontology/caf.owl#interest">
      <rdfs:domain rdf:resource="http://localhost/caf/ontology/caf.owl#User"/>
      <rdfs:range rdf:resource="http://localhost/caf/ontology/caf.owl#Context"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:about="http://xmlns.com/foaf/0.1/nick">
      <rdfs:domain rdf:resource="http://localhost/caf/ontology/caf.owl#User"/>
    </owl:ObjectProperty>
    <rdf:Property rdf:about="http://purl.org/dc/elements/1.1/title">
      <rdfs:domain rdf:resource="http://localhost/caf/ontology/caf.owl#Group"/>
    </rdf:Property>
    <rdf:Description rdf:about="http://localhost/caf/ontology/caf.owl#hasOwner">
      <rdf:type>
        <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#ObjectProperty"/>
      </rdf:type>
      <rdfs:domain>
        <rdf:Description
rdf:about="http://localhost/caf/ontology/caf.owl#Repository"/>
      </rdfs:domain>
      <rdfs:range>
        <rdf:Description rdf:about="http://localhost/caf/ontology/caf.owl#User"/>
      </rdfs:range>
    </rdf:Description>
    <rdf:Description rdf:about="http://localhost/caf/ontology/caf.owl#hasContent">
      <rdf:type>
        <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#ObjectProperty"/>
      </rdf:type>
      <rdfs:domain>
        <rdf:Description rdf:about="http://localhost/caf/ontology/fohm.owl#Data"/>
      </rdfs:domain>
      <rdfs:range>
        <rdf:Description rdf:about="http://xmlns.com/foaf/0.1/Document"/>
      </rdfs:range>
      <rdfs:subPropertyOf>
        <rdf:Description
rdf:about="http://localhost/caf/ontology/fohm.owl#hasRegionContent"/>
      </rdfs:subPropertyOf>
    </rdf:Description>
    <rdf:Description
rdf:about="http://localhost/caf/ontology/fohm.owl#hasRegionContent">
      <rdf:type>
        <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#ObjectProperty"/>
      </rdf:type>
    </rdf:Description>
    <rdf:Description rdf:about="http://xmlns.com/foaf/0.1/Document">
      <rdf:type>
```

```
      <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Class"/>
    </rdf:type>
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost/caf/ontology/caf.owl#inRepository">
    <rdf:type>
      <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    </rdf:type>
    <rdfs:domain>
      <rdf:Description rdf:about="http://localhost/caf/ontology/fohm.owl#Data"/>
    </rdfs:domain>
    <rdfs:range>
      <rdf:Description
rdf:about="http://localhost/caf/ontology/caf.owl#Repository"/>
    </rdfs:range>
  </rdf:Description>
</rdf:RDF>
<!-- Created with Protege (with OWL Plugin 3.2.1, Build 365)
http://protege.stanford.edu -->
```

FOHM ontology

```
<?xml version="1.0"?>
<rdf:RDF xml:base="file:/D:/AppServ/www/caf/ontology/fohm.owl"
xmlns="file:/D:/AppServ/www/caf/ontology/fohm.owl#"
xmlns:caf="http://localhost/caf/ontology/caf.owl#"
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:dcterms="http://purl.org/dc/terms/"
xmlns:fohm="http://localhost/caf/ontology/fohm.owl#"
xmlns:fohmrt="http://localhost/caf/ontology/fohmrt.rdf#"
xmlns:nav="http://localhost/caf/ontology/nav.owl#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:skos="http://www.w3.org/2004/02/skos/core#">
  <owl:Ontology rdf:about="file:/D:/AppServ/www/caf/ontology/fohm.owl"/>
  <owl:Class rdf:about="http://localhost/caf/ontology/fohm.owl#Data">
    <dc:creator>
      <owl:Class rdf:about="http://localhost/caf/ontology/caf.owl#User"/>
    </dc:creator>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="http://purl.org/dc/elements/1.1/creator"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
```

```
          <owl:maxCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:maxCardinality>
          <owl:onProperty>
            <owl:ObjectProperty rdf:about="http://purl.org/dc/terms/modified"/>
          </owl:onProperty>
        </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:maxCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:maxCardinality>
          <owl:onProperty>
            <owl:ObjectProperty rdf:about="http://purl.org/dc/elements/1.1/title"/>
          </owl:onProperty>
        </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:maxCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:maxCardinality>
          <owl:onProperty>
            <owl:ObjectProperty rdf:about="http://purl.org/dc/terms/created"/>
          </owl:onProperty>
        </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
        <owl:Class
rdf:about="http://localhost/caf/ontology/fohm.owl#ReferenceableObject"/>
      </rdfs:subClassOf>
    </owl:Class>
    <owl:Class rdf:about="http://localhost/caf/ontology/fohm.owl#Feature"/>
    <owl:Class rdf:about="http://localhost/caf/ontology/fohm.owl#Behaviour"/>
    <owl:Class rdf:about="http://localhost/caf/ontology/fohm.owl#ContextualObject"/>
    <owl:Class rdf:about="http://localhost/caf/ontology/fohm.owl#Reference">
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:onProperty>
            <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/fohm.owl#hasStart"/>
          </owl:onProperty>
          <owl:maxCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:maxCardinality>
        </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:onProperty>
            <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/fohm.owl#hasEnd"/>
          </owl:onProperty>
```

```
            <owl:maxCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:maxCardinality>
          </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf
rdf:resource="http://localhost/caf/ontology/fohm.owl#ContextualObject"/>
        <rdfs:subClassOf>
          <owl:Restriction>
            <owl:onProperty>
              <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/fohm.owl#hasReferencedObject"/>
            </owl:onProperty>
            <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
          </owl:Restriction>
        </rdfs:subClassOf>
      </owl:Class>
      <owl:Class rdf:about="http://localhost/caf/ontology/fohm.owl#Binding">
        <rdfs:subClassOf
rdf:resource="http://localhost/caf/ontology/fohm.owl#ContextualObject"/>
        <rdfs:subClassOf>
          <owl:Restriction>
            <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
            <owl:onProperty>
              <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/fohm.owl#hasReferenceBound"/>
            </owl:onProperty>
          </owl:Restriction>
        </rdfs:subClassOf>
      </owl:Class>
      <owl:Class rdf:about="http://www.w3.org/2004/02/skos/core#Concept"/>
      <owl:Class rdf:about="http://localhost/caf/ontology/fohm.owl#Relation"/>
      <owl:Class
rdf:about="http://localhost/caf/ontology/fohm.owl#ReferenceableObject">
        <rdfs:subClassOf
rdf:resource="http://localhost/caf/ontology/fohm.owl#ContextualObject"/>
      </owl:Class>
      <owl:Class rdf:about="http://localhost/caf/ontology/caf.owl#Context"/>
      <owl:Class rdf:about="http://localhost/caf/ontology/fohm.owl#LocSpec"/>
      <owl:Class rdf:about="http://localhost/caf/ontology/fohm.owl#Association">
        <rdfs:subClassOf
rdf:resource="http://localhost/caf/ontology/fohm.owl#ReferenceableObject"/>
        <rdfs:subClassOf>
          <owl:Restriction>
            <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
            <owl:onProperty>
              <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/fohm.owl#hasBinding"/>
            </owl:onProperty>
```

```
          </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
          <owl:Restriction>
            <owl:onProperty>
              <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/fohm.owl#hasDescription"/>
            </owl:onProperty>
            <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
          </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
          <owl:Restriction>
            <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
            <owl:onProperty>
              <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/fohm.owl#hasRelationType"/>
            </owl:onProperty>
          </owl:Restriction>
        </rdfs:subClassOf>
      </owl:Class>
      <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/fohm.owl#hasContext">
        <rdfs:domain
rdf:resource="http://localhost/caf/ontology/fohm.owl#ContextualObject"/>
        <rdfs:range rdf:resource="http://localhost/caf/ontology/caf.owl#Context"/>
      </owl:ObjectProperty>
      <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/fohm.owl#hasReferencedObject">
        <rdfs:domain rdf:resource="http://localhost/caf/ontology/fohm.owl#Reference"/>
        <rdfs:range
rdf:resource="http://localhost/caf/ontology/fohm.owl#ReferenceableObject"/>
      </owl:ObjectProperty>
      <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/fohm.owl#hasRelationType">
        <rdfs:domain
rdf:resource="http://localhost/caf/ontology/fohm.owl#Association"/>
        <rdfs:range rdf:resource="http://localhost/caf/ontology/fohm.owl#Relation"/>
      </owl:ObjectProperty>
      <owl:ObjectProperty rdf:about="http://localhost/caf/ontology/fohm.owl#hasStart">
        <rdfs:domain rdf:resource="http://localhost/caf/ontology/fohm.owl#LocSpec"/>
      </owl:ObjectProperty>
      <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/fohm.owl#hasDescription">
        <rdfs:domain
rdf:resource="http://localhost/caf/ontology/fohm.owl#Association"/>
      </owl:ObjectProperty>
      <owl:ObjectProperty rdf:about="http://purl.org/dc/elements/1.1/title">
        <rdfs:domain rdf:resource="http://localhost/caf/ontology/fohm.owl#Data"/>
```

```
    </owl:ObjectProperty>
    <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/fohm.owl#hasFeature">
        <rdfs:range rdf:resource="http://localhost/caf/ontology/fohm.owl#Feature"/>
        <rdfs:domain rdf:resource="http://localhost/caf/ontology/fohm.owl#Binding"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:about="http://purl.org/dc/terms/created">
        <rdfs:domain rdf:resource="http://localhost/caf/ontology/fohm.owl#Data"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/fohm.owl#hasBinding">
        <rdfs:range rdf:resource="http://localhost/caf/ontology/fohm.owl#Binding"/>
        <rdfs:domain
rdf:resource="http://localhost/caf/ontology/fohm.owl#Association"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:about="http://localhost/caf/ontology/fohm.owl#hasEnd">
        <rdfs:domain rdf:resource="http://localhost/caf/ontology/fohm.owl#LocSpec"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/fohm.owl#hasReferenceBound">
        <rdfs:range rdf:resource="http://localhost/caf/ontology/fohm.owl#Reference"/>
        <rdfs:domain rdf:resource="http://localhost/caf/ontology/fohm.owl#Binding"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:about="http://localhost/caf/ontology/caf.owl#hasData">
        <rdfs:domain rdf:resource="http://localhost/caf/ontology/fohm.owl#Reference"/>
        <rdfs:range>
            <rdf:Description rdf:about="http://localhost/caf/ontology/fohm.owl#Data"/>
        </rdfs:range>
    </owl:ObjectProperty>
    <owl:ObjectProperty
rdf:about="http://localhost/caf/ontology/fohm.owl#hasBehaviour">
        <rdfs:range rdf:resource="http://localhost/caf/ontology/fohm.owl#Behaviour"/>
        <rdfs:domain
rdf:resource="http://localhost/caf/ontology/fohm.owl#ContextualObject"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:about="http://purl.org/dc/terms/modified">
        <rdfs:domain rdf:resource="http://localhost/caf/ontology/fohm.owl#Data"/>
    </owl:ObjectProperty>
    <rdf:Description rdf:about="foaf:Document">
        <rdf:type>
            <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Class"/>
        </rdf:type>
    </rdf:Description>
    <rdf:Description
rdf:about="http://localhost/caf/ontology/fohm.owl#hasRegionContent">
        <rdf:type>
            <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#ObjectProperty"/>
        </rdf:type>
        <rdfs:domain>
            <rdf:Description rdf:about="http://localhost/caf/ontology/fohm.owl#Data"/>
```

```
    </rdfs:domain>
    <rdfs:range>
      <rdf:Description rdf:about="foaf:Document"/>
    </rdfs:range>
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost/caf/ontology/fohm.owl#hasLocSpec">
    <rdf:type>
      <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    </rdf:type>
    <rdfs:domain>
      <rdf:Description
rdf:about="http://localhost/caf/ontology/fohm.owl#Reference"/>
    </rdfs:domain>
    <rdfs:range>
      <rdf:Description
rdf:about="http://localhost/caf/ontology/fohm.owl#LocSpec"/>
    </rdfs:range>
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost/caf/ontology/caf.owl#hasReference">
    <rdf:type>
      <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    </rdf:type>
    <rdfs:domain>
      <rdf:Description rdf:about="http://localhost/caf/ontology/fohm.owl#Data"/>
    </rdfs:domain>
    <rdfs:range>
      <rdf:Description
rdf:about="http://localhost/caf/ontology/fohm.owl#Reference"/>
    </rdfs:range>
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost/caf/ontology/caf.owl#inRepository">
    <rdf:type>
      <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    </rdf:type>
    <rdfs:domain>
      <rdf:Description rdf:about="http://localhost/caf/ontology/fohm.owl#Data"/>
    </rdfs:domain>
    <rdfs:range>
      <rdf:Description
rdf:about="http://localhost/caf/ontology/caf.owl#Repository"/>
    </rdfs:range>
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost/caf/ontology/caf.owl#Repository">
    <rdf:type>
      <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Class"/>
    </rdf:type>
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost/caf/ontology/caf.owl#inAssociation">
```

```
        <rdf:type>
            <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#ObjectProperty"/>
        </rdf:type>
        <rdfs:domain>
            <rdf:Description
rdf:about="http://localhost/caf/ontology/fohm.owl#Association"/>
        </rdfs:domain>
        <rdfs:range>
            <rdf:Description rdf:about="http://localhost/caf/ontology/fohm.owl#Binding"/>
        </rdfs:range>
    </rdf:Description>
    <rdf:Description rdf:about="http://localhost/caf/ontology/caf.owl#inBinding">
        <rdf:type>
            <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#ObjectProperty"/>
        </rdf:type>
        <rdfs:domain>
            <rdf:Description
rdf:about="http://localhost/caf/ontology/fohm.owl#Reference"/>
        </rdfs:domain>
        <rdfs:range>
            <rdf:Description rdf:about="http://localhost/caf/ontology/fohm.owl#Binding"/>
        </rdfs:range>
    </rdf:Description>
</rdf:RDF>
<!-- Created with Protege (with OWL Plugin 3.2.1, Build 365)
http://protege.stanford.edu ->
```

Navigation Ontology

```
<?xml version="1.0"?>
<rdf:RDF xmlns:fohm="http://localhost/caf/ontology/fohm.owl#"
xmlns:nav="http://localhost/caf/ontology/nav.owl#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:skos="http://www.w3.org/2004/02/skos/core#">
    <rdf:Description rdf:about="nav:DirectionAssociation">
        <rdf:type>
            <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Class"/>
        </rdf:type>
        <rdfs:subClassOf>
            <rdf:Description rdf:about="fohm:Association"/>
        </rdfs:subClassOf>
    </rdf:Description>
    <rdf:Description rdf:about="nav:DirectionFeature">
        <rdf:type>
            <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Class"/>
        </rdf:type>
        <rdfs:subClassOf>
```

```
          <rdf:Description rdf:about="fohm:Feature"/>
        </rdfs:subClassOf>
        <owl:oneOf rdf:parseType="Collection">
          <rdf:Description
rdf:about="http://localhost/caf/ontology/nav.owl#Destination"/>
          <rdf:Description rdf:about="http://localhost/caf/ontology/nav.owl#Source"/>
        </owl:oneOf>
    </rdf:Description>
    <rdf:Description rdf:about="fohm:Feature">
        <rdf:type>
          <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Class"/>
        </rdf:type>
    </rdf:Description>
    <rdf:Description rdf:about="fohm:hasFeature">
        <rdf:type>
          <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#ObjectProperty"/>
        </rdf:type>
    </rdf:Description>
    <rdf:Description rdf:about="nav:hasDirectionFeature">
        <rdf:type>
          <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#ObjectProperty"/>
        </rdf:type>
        <rdfs:domain>
          <rdf:Description rdf:about="nav:DirectionAssociation"/>
        </rdfs:domain>
        <rdfs:range>
          <rdf:Description rdf:about="nav:DirectionFeature"/>
        </rdfs:range>
        <rdfs:subPropertyOf>
          <rdf:Description rdf:about="fohm:hasFeature"/>
        </rdfs:subPropertyOf>
    </rdf:Description>
    <rdf:Description rdf:about="fohm:Association">
        <rdf:type>
          <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Class"/>
        </rdf:type>
    </rdf:Description>
    <rdf:Description rdf:about="fohm:hasBinding">
        <rdf:type>
          <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#ObjectProperty"/>
        </rdf:type>
    </rdf:Description>
    <rdf:Description>
        <rdf:type>
          <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#AllDifferent"/>
        </rdf:type>
        <owl:distinctMembers rdf:parseType="Collection">
```

```
          <rdf:Description
rdf:about="http://localhost/caf/ontology/nav.owl#Destination">
            <skos:definition>Destination</skos:definition>
            <skos:prefLabel>Destination</skos:prefLabel>
          </rdf:Description>
          <rdf:Description rdf:about="http://localhost/caf/ontology/nav.owl#Source">
            <skos:definition>Source</skos:definition>
            <skos:prefLabel>Source</skos:prefLabel>
          </rdf:Description>
        </owl:distinctMembers>
    </rdf:Description>
    <rdf:Description
rdf:about="http://localhost/caf/ontology/nav.owl#hasReferenceBound">
        <rdf:type>
          <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#ObjectProperty"/>
        </rdf:type>
        <rdfs:domain>
          <rdf:Description rdf:about="http://localhost/caf/ontology/fohm.owl#Binding"/>
        </rdfs:domain>
        <rdfs:range>
          <rdf:Description
rdf:about="http://localhost/caf/ontology/fohm.owl#Reference"/>
        </rdfs:range>
    </rdf:Description>
    <rdf:Description rdf:about="http://localhost/caf/ontology/fohm.owl#Binding">
        <rdf:type>
          <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Class"/>
        </rdf:type>
    </rdf:Description>
    <rdf:Description rdf:about="http://localhost/caf/ontology/fohm.owl#Reference">
        <rdf:type>
          <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Class"/>
        </rdf:type>
    </rdf:Description>
</rdf:RDF
```