A SERVICE-ORIENTED ARCHITECTURE FOR ADAPTIVE AND COLLABORATIVE E-LEARNING SYSTEMS

Maram Meccawy, MSc, BSc

Thesis submitted to the University of Nottingham
for the degree of Doctor of Philosophy

September 2008

Dedication

To my parents

ABSTRACT

This research proposes a new architecture for Adaptive Educational Hypermedia Systems (AEHS). Architectures in the context of this thesis refer to the components of the system and their communications and interactions. The architecture addresses the limitations of AEHS regarding interoperability, reusability, openness, flexibility, and limited tools for collaborative and social learning. It presents an integrated adaptive and collaborative Web-based learning environment.

The new e-learning environment is implemented as a set of independent Web services within a service-oriented architecture (SOA). Moreover, it uses a modern Learning Management System (LMS) as the delivery service and the user interface for this environment. This is a two-way solution, whereby adaptive learning is introduced via a widely adopted LMS, and the LMS itself is enriched with an external - yet integrated - adaptation layer.

To test the relevance of the new architecture, practical experiments were undertaken. The interoperability, reusability and openness test revealed that the user could easily switch between various LMS to access the personalised lessons. In addition, the system was tested by students at the University of Nottingham as a revision guide to a Software Engineering module. This test showed that the system was robust; it automatically handled a large number of students and produced the desired adaptive content. However, regarding the use of the collaborative learning tools, the test showed low levels of such usage.

ACKNOWLEDGMENTS

Firstly, all thanks to Allah, The Most Gracious, Most Merciful, for all the blessings that I continue to receive and for the strength that He gave me to complete this PhD.

Secondly, I would like to express my gratitude to those who have supported me during this challenging journey of research for without their presence in my life reaching my destination would have been more difficult.

I start by thanking my parents: Dr. Abdulrahman Meccawy and Mrs. Hayat Hemdi; thanks Mum for your love, support and prayers, and thanks Dad for your encouragement and for believing in me all the way.

Also, many thanks to my supervisors, Dr. Peter Blanchfield, who although had agreed to supervise me rather late in this journey, yet his continuous advise and support helped me to move forward and achieve my research goals. In addition, I would like to thank Dr. Helen Ashman and Dr. Tim Brailsford, whom I have begun this research with for many valuable discussions.

Moreover, I would like to thank those who have helped during the initial proof-reading stage. The list includes: my best friend Eva Martinez, who had spent a lot of her precious time reading many of my chapters, my sister Zilal Meccawy as well as my friends: Marta Ray-Lopez, Hala Obiedat and Asni Taher. Thanks girls, your comments and feedback were invaluable and much appreciated.

Last but not least, I want to express my gratitude to the Ministry of Higher Education in Saudi Arabia for their financial support. This also includes the Saudi Cultural Bureau in London, especially Dr. Mohammed Diaa, who has offered me all the possible help, advice and motivation even when he ceased to be my supervisor.

CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF PUBLICATIONS[1]

1. Meccawy, M., and Blanchfield, P. Combining Adaptive and Collaborative Learning: A Case Study. In: Proc. IGACLE: Individual and Group Adaptation in Collaborative Learning Environments Workshop at EC-TEL 08: 3rd European Conference on Technology-Enhanced Learning. Maastricht, The Netherlands, September 16-19. 2008.

2. Meccawy, M., Blanchfield, P., Ashman, H., Brailsford, T., and Moore, A. WHURLE 2.0: Adaptive Learning Meets Web 2.0. In: Proc. EC-TEL 08: 3rd European Conference on Technology-Enhanced Learning Maastricht, The Netherlands: Springer Verlag, September 16-19. 2008.

3. Meccawy, M., Brusilovsky, P., Ashman, H., Yudelson, M. and Scherbinina, O. Integrating Interactive Learning Content into an Adaptive E-Learning System: Lessons Learned. In: Proc. World Conference on E-Learning in Corporate, Government, Health Care, & Higher Education (e-Learn 2007). Quebec City, Canada Oct. 15-19. 2007.

4. Rey López, M., Brusilovsky, P., Meccawy, M., Díaz Redondo, R., Fernández Vilas, A., and Ashman, H. Resolving the Problem of Intelligent Learning Content in Learning Management Systems. *International Journal on E-Learning (IJEL)*. 2007. 7 (3).

5. Meccawy, M., Stewart, C., and Ashman, H., Adaptive Educational Hypermedia Interoperability and Content Creation with a Web Service-based Architecture. *International Journal of Learning Technology (IJLT)*.2007. 7 (3): p. 269 - 285.

6. Meccawy, M. WHURLE 2.0: An Adaptive Web Service e-Learning Environment for the Web. In: Proc. Saudi Innovation Conference (SIC 2007). Newcastle upon Tyne, UK May 11-12. 2007.

7. Meccawy, M., Stewart, C., and Ashman, H. Adaptive Educational Hypermedia Content Creation: A Web Service Based Architecture In: Proc. Adaptive Hypermedia (Ah2006) Workshop on Authoring for Adaptive Hypermedia. Dublin, Ireland, June 20. 2006.

8. Meccawy, M., Celik, I., Cristea, A., Stewart, C., and Ashman, H. Interoperable Adaptive Educational Hypermedia: A Web Service Definition. In: Proc. 6th IEEE International Conference on Advanced Learning Technologies (ICALT 2006). Kerkrade, The Netherlands: IEEE Computer Society, July 5-7. 2006.

[1] All the publications are available from the Website: http://www.cs.nott.ac.uk/~mzm/Publication.html

NOTE

In this thesis some abbreviations have been used to refer to both the singular and plural noun(s). Examples are listed below.

- **LMS** represents both Learning Management Systems and a Leaning Management System.
- **AEHS** stands for Adaptive Educational Hypermedia Systems and an Adaptive Educational Hypermedia System. In addition, **AEH** stands for Adaptive Educational Hypermedia.
- **SOA** is the abbreviation for Service-Oriented Architectures or a Service-Oriented Architecture.

CHAPTER 1

“The beginning of knowledge is the discovery of something we do not understand”

Frank Herbert (1920 - 1986)

INTRODUCTION

1.1 Overview

In its relatively short existence, the Web has had an unprecedented impact on our lives. The many barriers that it removed and the almost endless opportunities that it has brought are inarguable.

‘One-size-fits-all’, was the case in those early days when the Web was in its infancy. Given the technology at that time, the Web’s content, as well as its navigational paths, were almost identical to all users. As the Web started to mature and the number of its users increased and became more diverse, this unified approach faced criticism. Web users are unique individuals each with their own backgrounds, experiences, cultures, goals, abilities or interests. Moreover, they could be browsing using different devices. Hence the search began to identify a Web that follows a more flexible approach; a Web that adapts or an *Adaptive Web*. In this Web the users’ data is collected to build a model in order to provide a tailored or personalised Web experience. Personalisation and adaptation on the Web attracted wider attention from researches and new applications such as recommender systems became available to end users.

In the field of e-learning, seen as one of the pillars of the modern knowledge-based societies, the same question regarding personalisation and adaptation was asked: is personalised learning needed to accommodate students with different knowledge, experiences, goals or learning styles? Realising the uniqueness of each individual student has resulted in an answer that is in favour of adaptation (Brusilovsky 2007a). This chapter provides an introduction to the research presented in this thesis which is placed in the field of adaptive and personalised Web-based learning.

1.2 Research Motivation and Background Overview

Adaptive Educational Hypermedia Systems (AEHS) (Brusilovsky 2001a) adapt to different models to provide a personalised learning experience for each student. A number of studies have suggested their value for online learning (Boyle and Encarnacion 1994; Eklund and Brusilovsky 1998; Brusilovsky 1998 ; Conlan and Wade 2004 b). With Adaptive Educational Hypermedia (AEH), online learning is able to achieve the tutors' or learners' goals with techniques that are difficult to apply in a traditional classroom with a large number of students due to time, cost and effort constraints. It is cumbersome for a teacher to repeat each lesson to suit the needs of every student. For example, how could he/she hide or show information from or to a particular student? How could the same lesson be presented at the same time to accommodate different learning styles? However, it is possible to implement such learning scenarios through adaptive hypermedia systems. Moreover, the significant impact of personalised learning can be seen in the case of Web-based distance learning courses where the absence of the teacher/tutor advocates the need of some sort of guidance for the student to assist his/her learning needs.

AEH can be seen as a user-centred design approach in e-learning that is driven by the needs of the users (learners, tutors or both) and not by the technology. User-centred methodologies address issues of quality, usability and evaluation of systems by ensuring that the users are considered at all stages of the course's development (Hussein 2005). However, the poor presence of adaptive systems in real world e-learning has encouraged researchers to investigate its own problems and limitations such as in (Brusilovsky 2004b). This shows that the majority of AEHS are not designed for the modern e-learning context which prompts services and reusability of learning content (Brusilovsky 2008). In terms of easy user-level content authoring tools and course administration, they are usually limited and require in-depth technical knowledge lacked by the majority of non-IT teachers (Stewart *et al.* 2005). Any teacher who is considering authoring an adaptive lesson need to decide which AEHS should he/she use? Then he/she may select a specific system to learn how to use and spend time creating materials for this system. But if the chosen system stops being developed or maintained, the author can either continue to develop content for a system that will eventually go out of date, or can move to a new system that the previously created content is likely to be incompatible with. This means a whole new cycle of learning and content-development which results in a huge loss of resources.

Since the authoring for such systems remains expensive, complex and time consuming, reusability of this content becomes essential. However, the majority of early AEHS were built as monolithic systems that cannot share their content to allow reusability, or are limited in doing so. Nor can they share user profiles, adaptation rules, or any other AEHS components, hence each system has to be used as a whole, even if the interest concerns only one function or component (Brusilovsky 2008). These limitations have encouraged different research groups to investigate ways to provide more flexible and interoperable adaptive systems. Interoperability here is the ability of a system or a product to work with other differt systems or products (inter-operate). Extensive research has been done on interoperability and usability of adaptive learning content, which includes using different approaches, some of which use standards, others provide new research solutions (Rey López *et al.* 2007).

Research exists on generating a new authoring paradigm, one that moves away from the ‘create once, use once’ approach to the ‘create once, use many’ approach (Stewart *et al.* 2005). It has stated that only interoperability between AEHS could address these authoring issues. When an author is able to develop learning materials in one system and then have them delivered in another, AEHS would begin to enter the mainstream of education.

Another key issue with AEHS is that they often have very limited social and collaborative learning activities although learning is believed to be a social activity according to Bandura’s *Social Learning Theory* (1977) as cited in (Tu 2000).

Providing a one-purpose adaptive system that isolates the learners would not be able to compete with other e-learning systems such Learning Management Systems (LMS), which are currently dominating Web-based learning. LMS are equipped with a number of essential tools and activities that take into account different stakeholders in the e-learning processes: administrators, teachers and students, and hence provide a general purpose and easy to use learning environment with appealing user-interfaces.

In recent years new paradigms have emerged on the Web whereby communities rather than documents are paramount. This new approach to hypermedia, which has been referred to as *Web 2.0*, focuses on services rather than software (O'Reilly 2006). It has prompted the redesign of an online educational system to become a complete learning environment where students can socialise and interact. The online learning process is transformed from one that is considered to provide a passive learning experience where students only receive information from the system without being able to interact or contribute, into one where they learn by doing, interacting and socialising. This new generation of e-Learning which is evolving with the new generation of the Web and the Internet as a whole is known as *e-Learning 2.0* (Downes 2005). Modern LMS are designed to provide these types of interactive and social e-learning environments (Millard and Essex 2007)

The success of LMS and the poor presence of AEHS have driven some researchers in Adaptive Educational Hypermedia (AEH) to reconsider the designs of their systems. They believe that the problem for AEHS arises from their architectural design failing to answer the overall needs of modern Web-enhanced learning rather than the actual adaptation and personalisation functionality failing to perform (Brusilovsky 2008). This thesis presents work which focuses on this research area as will be explained in detail in the next section.

1.3 Problem Under Study

The aforementioned state of AEHS leads to the conclusion that their major drawback appears to be their architectures (Brusilovsky 2004b), implementations or delivery systems failing to answer the needs of today's e-Learning requirements. The research presented in this thesis shares this belief and hence reconsiders such designs of AEHS in the light of the current needs of learners and tutors as well as the advances in Web technologies. Such technologies include *Web services* (WSG 2004), which are sets of technologies and standards that are designed to allow for more interoperability and flexibility between Web-based distributed applications.

The focus of the research presented here is on how to improve these architectures and enable adaptation delivery through a reliable, flexible, and collaborative learning environment, which can be widely adopted in the e-learning community. Therefore, these improvements would eventually help in lifting AEHS from their current undesirable position and allow them to penetrate into the main stream e-learning.

Would the new architecture work? And would it address the current limitations of Adaptive Educational Hypermedia Systems and satisfy the requirements of modern Web-based learning environments?

Research presented in this thesis investigates these questions in order to provide a sustainable solution for adaptive Web-based learning.

1.4 Aims, Objectives and Methodologies

1.4.1 Aims

The overall aim of this research is to promote a wider use of adaptive e-learning by improving the architectures of Adaptive Educational Hypermedia Systems, overcoming limitations of the existing systems. Such shortcomings include: limited interoperability and reusability between different adaptive systems especially when compared with what is provided through using a Learning Management System. In addition, the research endeavours to provide an enhanced e-learning environment that satisfies the requirements of the new generation Web and its users. This is achieved by combining personalised and adaptive learning with collaborative learning concepts and tools, and presenting learning content through appealing (non-prototypic) user-interface.

1.4.2 Objectives

1. To identify the problems which are limiting the use of Adaptive Educational Hypermedia Systems, and to analyse requirements of the new generation of e-learning systems.
2. To investigate the problem of interoperability and reusability of adaptive and interactive e-learning content.
3. To design an architecture for Web-based learning that has adaptation as one of its core features; this architecture also provides an environment for collaborative learning.
4. To develop and implement this architecture.
5. To test the implemented architecture for its interoperability, reusability, flexibility and openness, robustness and reliability and collaborative learning features.

1.4.3 Methodologies

Throughout this research different methods are used to collect data and test the conceptual design. These methods include analysis of the literature, investigations of the issue of interoperability of content through case studies, development of a new architecture, and applying technical experimental tests to the developed system. In addition, a case study for users' trials which also involved an online questionnaire is designed to test the system's validity, reliability and monitor the students' use of social tools.

1.5 Thesis Contributions

This thesis presents an innovative Web service-oriented architecture for an adaptive e-learning framework. It is designed to address the aforementioned limitations encountered in former adaptive Educational Hypermedia Systems. Moreover, takes into consideration the current advances of the Web in addition to the demands of modern Web-based education. The architecture's flexibility allows for the adaptive content to be delivered in any Learning Management System, which is used as delivery services (portal) for this integrated e-learning environment. This results in an integrated Web-based learning environment that is enhanced with both adaptation and social learning. The new architecture is known as WHURLE 2.0.

Since limited interoperability is a significant challenge in AEHS, different approaches have been proposed in the literature to address this issue. As a result of examining these approaches a new classification that groups those proposals into four different categories is introduced by this thesis. The four categories are: the *Conversion* approach, the *Lingua-franca* approach, the *Universal language* approach and the *Languages-specifier* approach. Details and definitions of those four categories can be found in chapter 2 (section 2.5).

The research described here investigates (through case studies) two different methods for achieving interoperability and reusability of learning content in adaptive systems. The first case study explains work that extended the conversion approach by applying Web services technologies. It suggests that Web services, when applied to the conversion approach, resulted in the conversion program being more flexible and accessible, leading to increasing the potential for personalisation and adaptive learning. On the other hand, the second case study explores the reusability of rich adaptive or interactive content using one adaptive system's original protocols. It shows practically that the interoperability of rich, intelligent or adaptive content is still a problem that needs to be resolved to allow different LMS to launch this type of learning content. In addition, this study highlighted that the use of proprietary protocols might not be feasible solution for interoperability.

Finally, the evaluation case study reveals some interesting findings regarding students' collaboration online; the results show a low degree of interaction among students using the available Web 2.0 or collaboration tools such as Blogs or Wikis.

To date, 8 peer-reviewed publications have arisen from this research (See page XII).

1.6 Thesis Structure

The thesis is composed of nine chapters. Chapter 2 describes in greater detail Adaptive Educational Hypermedia, its advantages, limitations and the current state of the art in terms of research that aims to address those limitations using a number of approaches. Chapter 3 focuses on another important aspect of e-learning which is almost absent from AEHS - that is, collaborative and social learning - and analyses how this is achieved by LMS and how the AEH community recognises this shortcoming and what proposals are made to tackle it.

Chapter 4 discusses the technical aspects of the architecture proposed in this thesis and justifies the use of certain technologies as its platform. Chapter 5 presents two case studies where interoperability of learning content in the context of AEHS has been explored, highlighting the learned lessons from each case. In chapter 6, the proposed learning environment with its service-oriented architecture in comparison with the previous monolithic system, from which this system evolved, is described. After that, the development of this new architectural design is reported in chapter 7. The evaluation process is then presented in chapter 8 through a definition of an evaluation framework. Lastly, chapter 9 discusses different aspects of this research focusing on the newly designed architecture. The chapter ends with conclusions and proposals for future work.

1.7 Summary

Personalised solutions in e-learning are needed to cater for students representing the realities of a wide spectrum of online users. A task that might be difficult to achieve through traditional classroom settings by teachers, could be well within the great capabilities of Web-based learning. Here the added values of e-learning are recognised, where the aim is to not just to deliver learning content online, but to improve the overall learning experiences. Adaptive learning as provided by Adaptive Educational Hypermedia aims to serve this need. However, its own limitations have not allowed it to take the lead or even become a recognised competitor to its rivals such as Learning Management Systems. On the other hand, the lack of adaptation is also limiting the LMS online capabilities and experiences.

Recent developments in the Web are focusing more on the interaction, communication and sharing between its users. This applies to online learning, which is one of the Web's rapidly developing applications. Therefore, those requirements were taken into consideration when designing a new e-learning environment that promotes both adaptation and social learning. This environment is built using a service-oriented approach that applies Web services technologies.

The next chapter presents online adaptation in the context of e-learning through a detailed overview of the past, present and future of Adaptive Educational Hypermedia.

CHAPTER 2

"If I have seen further, it is by standing on the shoulders of giants"
Isaac Newton (1642 - 1727)

ADAPTIVE EDUCATIONAL HYPERMEDIA: OVERVIEW AND CURRENT CHALLENGES

2.1 Overview

For most people their main interaction with a hypermedia system is their daily use of the World Wide Web which has given users the experience of navigating freely between decentralised networks of nodes of information connected by different links. Although this freedom of mobility is one of the Web's key advantages, it might not be, in some cases, the best it can offer to everyone.

A problem concerning current hypermedia systems, and the Web in particular, is the huge number of users with different interests, goals, preferences and needs, who are presented with the same information (content), navigation paths or presentation styles. Moreover, the number of users who browse using different devices is growing, and each of those devices has its own requirements, for example in terms of interfaces and resolution. One of the limitations of the traditional *static* hypermedia applications is that they provide almost the same surfing experience to all users, which could be a problem if the population is relatively diverse as in the case of the World Wide Web (Brusilovsky 2001b).

Adaptive Hypermedia (AH) is an alternative to traditional hypermedia systems that aims to overcome those problems by providing some help and guidance to users. In an educational context this is achieved through *Adaptive Educational Hypermedia* (AEH), which is where the research presented in this thesis is placed. AEH treats learners as distinct individuals that require personalised assistance and guidance during their online learning process. Despite the benefits and great potential of AEH, its implemented systems suffer from disadvantages that have limited their usage.

This chapter provides an overview of Adaptive Hypermedia and its importance in e-learning through Adaptive Educational Hypermedia Systems (AEHS). It investigates its advances as well as its current challenges that have resulted in its poor presence in the e-learning community. In addition, the chapter explores different approaches and proposals that try to address these challenges.

2.2 Adaptive Hypermedia

Adaptive hypermedia provides tailored content and navigation paths by adapting to the *user model,* which is a model that stores the distinct feature of the user (section 2.2.1). AH is a direction of research that can be traced back to the early 1990s which originated from research on both hypermedia and user modelling and aims to improve the usability of hypermedia. While classical hypermedia delivers the same page to every user, adaptive hypermedia stores the user's goal, preferences or knowledge and presents different pages to different users making it possible to deliver modified personalised views of a hypermedia document (Brusilovsky 1999).

Although research on this field started a few years before the great breakthrough of the Internet in the 1990s, the emergence of the Web led to a rapid growth of research activity on adaptive hypermedia. Brusilovsky defines Adaptive Hypermedia (AH) as:

> "*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*" (Brusilovsky 1996).

According to the previous definition, a system must have three characteristics to be considered an adaptive hypermedia system: first, to be a hypermedia system, second, to have a user model and third, to provide a distinct hypermedia view to a specific user (or group of users) based on this model. This is demonstrated by the following figure (2.1):

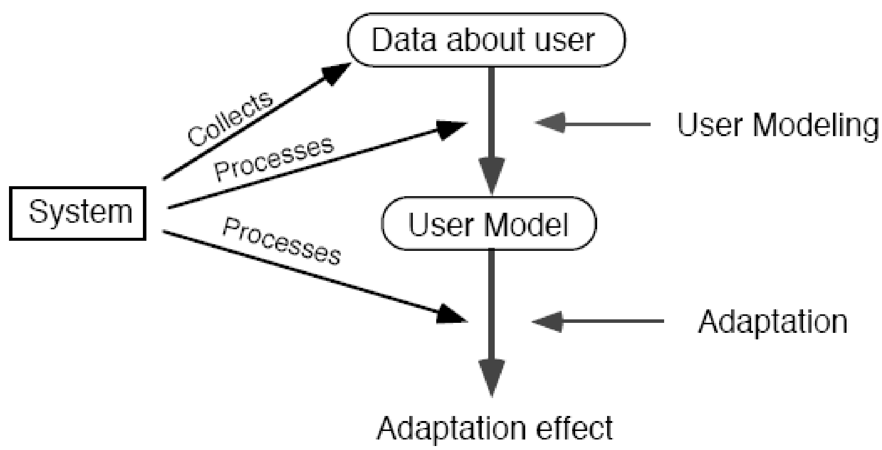

Figure 2.1: *"Classic loop 'user modeling – adaptation' in adaptive systems"*, taken from (Brusilovsky 1996).

As can be concluded from this figure, the system collects some specific data about the user either explicitly (for example by asking them to fill a questionnaire) or implicitly (by observing their interactions with the system). The system then processes the collected information and represents it as a model that is stored to be updated later as the user progresses or continues to use the system. Finally, the model is processed using an *adaptation engine* or *inference engine* and then the adaptation effect will take place.

Before moving further in AH, a distinction between two types of adaptive systems needs to be highlighted. Those two types are *adaptive* systems and *adaptable* systems. Adaptive systems are designed to adapt automatically without the need for the user's input, while an adaptable system provides the user with tools used to specify exactly how the system should be altered, allowing for more control to be delegated to the user (Oppermann 1994; Fumero 2006).

Research on adaptive hypermedia is taking place in many different Web applications such as e-learning, e-commerce and e-stores, e-museums, e-tourisms, e-media (news), e-health and others. Six types of adaptive hypermedia systems have been identified in (Mette and Thomas 1994): educational hypermedia, on-line information systems, on-line help systems, information retrieval hypermedia, institutional hypermedia and systems that manage personalised views in the information space. The application of AH in education is the field that is most relevant to the goals and objectives of this thesis and thus will receive greater attention.

When considering adaptation, there are important questions to be asked: 1) adapting to what? 2) What could be adapted? And 3) how is this adaptation effect presented? The following sections look for answers to these questions.

2.2.1 User Modelling

As has been mentioned earlier in this section, in order for adaptation to take place, information about the users should be stored in a user profile. Having the right information about the user is the heart of any adaptive hypermedia system. It is a core element in order to achieve any kind of adaptivity. A *user model* is the user's state of mind as represented by the adaptive system (De Bra 2004). *User modelling* is the process that results in having a user model. Each user will have a unique user profile that reflects a certain user's characteristics according to user model of that specific system. Therefore, a user model can also be defined as:

> *"a well-organized database which comprises information about the user and guides the system's inference engine"* (Mylonakis and Cullough 2003) .

A user's profile is aimed not to be static; in fact, it should be dynamically updated according to the user's interaction with the system to guide the adaptation engine.

Different researchers have chosen to make their system adapt to different aspects such as user's data or characteristics. In other systems, the adaptation has been built according to the usage data and environmental data. User's data is about the different characteristics of a user such as his/her interest or goals. As for the goals, they can be local, aiming, for example, to solve a specific task, or global, set towards long term objectives such as passing a course. Moreover, in the context of e-learning, for example, user's data can include his/her knowledge level about the subject domain, learning styles, cognitive styles (the preferred way for an individual to process information e.g.: analytical) or backgrounds (profession, language, prospect, capabilities or culture) (Santos *et al.* 2003). In this thesis, the focus is on the user's knowledge level, which has been implemented in the user modelling service (chapters 6 & 7).

On the other hand, usage data captures data about the users' interaction with the system such as their navigation history, while environmental data captures all aspects of users' information that are not specifically related to the user, such as their location or platform, for example, or the network's bandwidth, hardware, software and so forth (Paramythis and Loidl-Reisinger 2004). Hence, the difference between adaptation and personalisation could be distinguished as the former including adapting to all three types of adaptation (user, usage, environment), while personalisation focuses on user's data or needs (Graf and List 2005).

There are many types (or techniques) of user models, two of the well-known models being the *overlay* model and the *stereotype* model. The overlay model (Mette and Thomas 1994), is based on the structural model of the subject domain. It is a widely-used technique for user modelling which measures the user's knowledge level in any given topic or domain. In other words, an overlay model is a selected subset of the original domain model (Mylonakis and Cullough 2003). On the other hand, the stereotype model customises the knowledge for a specific group and then assigns a user to one group in which this user shares the same background, knowledge, goals or any other 'shared' data. It is a collection or cluster of user traits (Mette and Thomas 1994). Users cannot change from one group to another until their model has been updated according to their interaction with the system, resulting in their previous state changing so that it now satisfies the conditions for another group. In addition, a *hybrid* user model, which is a combination of both the overlay and the stereotype models, has been introduced in (Zakaria *et al.* 2002). Examples of two of the well-known user models' implementations include: CUMULATE (Brusilovsky 2003a) and Personis (Kay 2002).

2.2.2 Adaptation Types

Hypermedia consists primarily of nodes (content pages) and links. In Adaptive Hypermedia, the links, the pages (content) or both are adapted or personalised to suit an individual's (or a small group's) requirements. This introduces two types of adaptation technologies that deliver the adaptation or personalisation effect to the user; those are: *adaptive presentation* and *adaptive navigation support* (Brusilovsky 2001b). Adaptive presentation is adapting the content (information) of the page, for example, an expert user is provided with more detailed information than is a novice user or *vice versa*. Adaptive navigational support is about adapting the way the links are presented to the user on each webpage, as well as the navigational freedom or guidance through such links. It aims to help the user find an optimal path through a given hyperspace (Mylonakis and Cullough 2003).

Different types or categories of adaptation are presented in (Paramythis and Loidl-Reisinger 2004), which in addition to those mentioned above lists another category: the *adaptive interaction type* (adaptive interface), which is adaptation that takes place at the system's interface, to facilitate or support the user's interaction with the system without modifying the learning content.

2.2.3 Adaptation Methods and Techniques

Before presenting different methods and techniques for each type of adaptation, some terminology is worth clarifying. *Adaptation Techniques* are defined as**:**

> *"…methods of providing adaptation in existing AH systems. These techniques are a part of the implementation level of an AH system. Each technique can be characterized by a specific kind of knowledge*

representation and by a specific adaptation algorithm" (Brusilovsky 1996)

Adaptation methods are defined as**:**

> *"…generalization of existing adaptation techniques. Each method is based on a clear adaptation idea which can be presented at the conceptual level."* (Brusilovsky 1996)

When applying both the adaptation methods and techniques to the two main categories or types of adaptation, the following results are obtained:

- Adaptive Presentation Technique examples are conditional text, stretch-text, fragment variants and page variants techniques, frame-based techniques, and a combination of stretch-text and frame-based adaptation.
- Adaptive Presentation Method examples include: additional (prerequisite or comparative) explanations, explanation variants or sorting.
- Adaptive Navigation Support Techniques include: direct guidance, restricting access, sorting links, removing, disabling or hiding links, annotation and map adaptation.
- Adaptive Navigation Support Methods examples are: guidance methods (global and local) or orientation support methods (global and local).

The following figure (2.2) provides an overview of some of those adaptive hypermedia technologies:

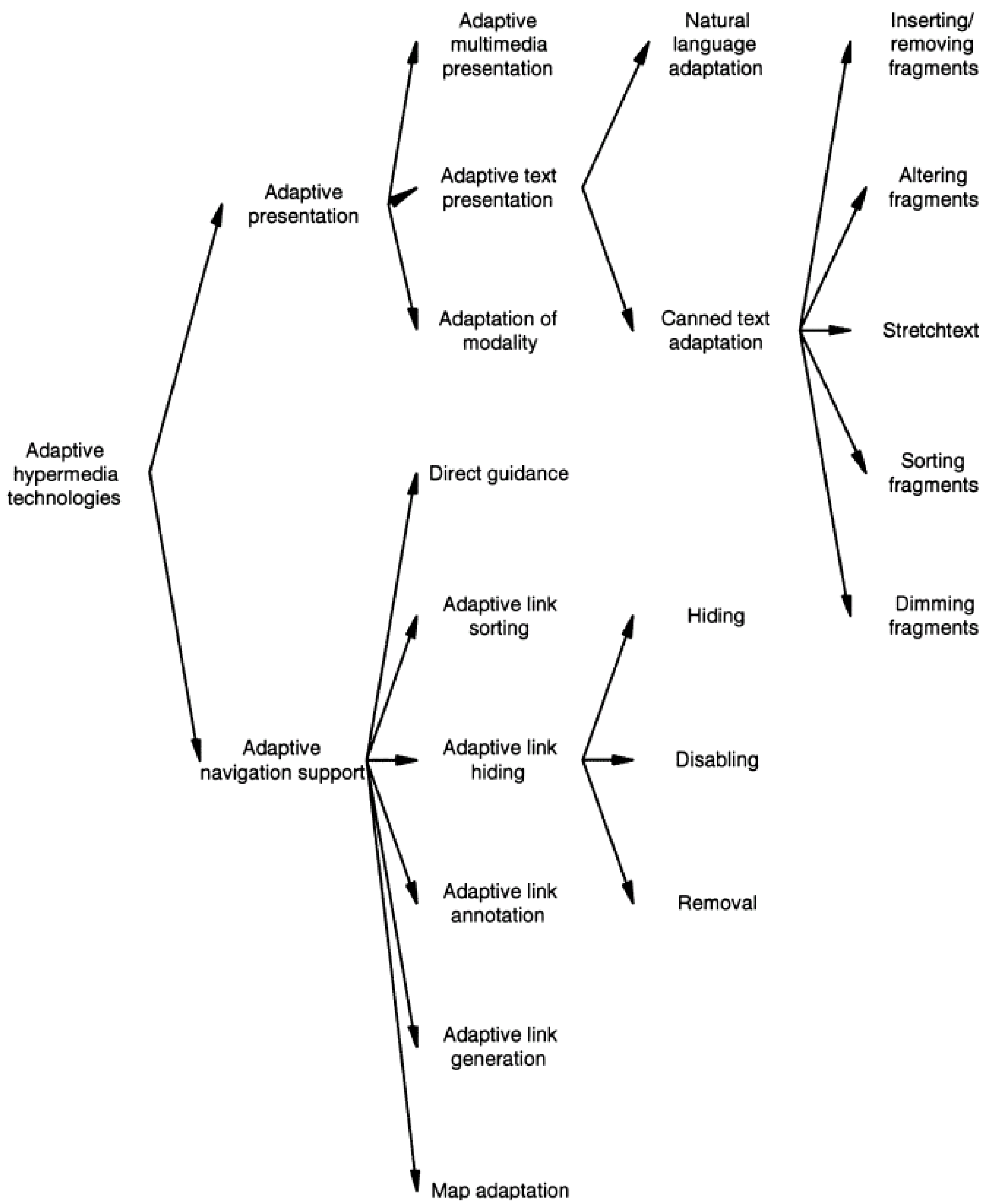

Figure 2.2: "The updated taxonomy of adaptive hypermedia technologies", taken from (Brusilovsky 2001b).

2.2.4 Disadvantages of Adaptive Hypermedia

Despite all the interesting features of adaptive hypermedia systems (Brusilovsky 1999), and the promising future that has been predicted regarding its applications and development (De Bra 2004), AH has its own limitations and there are some issues surrounding even the evaluation of these systems (De Bra 2000). The evaluations of AH systems have been criticised because most of them were performed to justify the use of adaptive hypermedia (AH) in a give application. This is done usually by comparing adaptive and non-adaptive versions of the same system. However, the evaluation is performed under testing conditions that favour the adaptive version.

One of AH's limitations is that the user is sometimes presented with irrelevant information through misguidance. Following a bad advice might result in worse performance than trying your best without getting any guidance. Moreover, the adaptation effects can confuse the users instead of accommodating their needs. For example, the removal of some links or sections after a certain number of visits can cause confusion or dissatisfaction. Therefore, it is advised to have a history option that will allow the user to track different versions of the page; this might be similar to what is provided by *wikipedia*[2] for instance.

2.4.5 Open Adaptive Hypermedia Systems

Open Adaptive Hypermedia Systems are systems that can respond to the individual needs regardless of where the users are, or where the content resides and what characteristics they bring in an open environment such as the Web. They use documents that belong to open databases or repositories of content (learning materials) and they are

[2] http://www.wikipedia.org/

open to all users (Mylonakis and Cullough 2003). The difference between open and closed adaptive hypermedia systems can be seen in the following definitions:

> ***"Definition 1 (Closed Corpus Adaptive Hypermedia System)***
>
> *A closed corpus adaptive hypermedia system is an adaptive hypermedia system which operates on a closed corpus of documents, where documents and relationships between the documents are known to the system at design time.*
>
> ***Definition 2 (Open Corpus Adaptive Hypermedia System)***
>
> *An open corpus adaptive hypermedia system is an adaptive hypermedia system which operates on an open corpus of documents, e.g., a set of documents that is not known at design time and, moreover, can constantly change and expand…"* (Brusilovsky 2007).

In the current state of AH, openness is considered as one of the key requirements for adaptive systems, as will be seen in the context of adaptive learning in the coming sections. This thesis presents an approach to AH that although it operates on a closed corpus, it aims to move forward towards such openness and flexibility by removing unnecessary boundaries and obstacles at least on a technical level.

2.3 Adaptive Educational Hypermedia Systems (AEHS), An Overview

Adaptive Educational Hypermedia Systems (AEHS) aim to overcome the lack of interpersonal interaction in online and distance learning by providing tailored learning experiences. The challenge is to make the system know and understand the learner's capabilities or goals and thus make the adaptation decisions instantly. This results in the complexity of the design, authoring and implementation of adaptive educational hypermedia systems in comparison with traditional Web-based educational systems. It is the price for getting the adaptation effects (Brusilovsky 2003b). The research described in this thesis lies in this area of adaptive Web-based learning and hence a greater emphasis is given to this application of AH from this point onward.

In adaptive educational hypermedia, the domain of a learning application can be represented as a hierarchy of concepts. The user model stores values for each concept for each user in the profile. In some cases it assumes that when a student logs into a page and reads it or passes an online test, he/she has learned the topic and the user profile is updated accordingly. Those applications also use a set of prerequisite relationships between these concepts, which together with the concept hierarchy and the concept values, allow the system to decide if the learner is ready for the new concept. For the links that are presented on each page, the system can also let a user know whether it leads to a page containing knowledge that is already known, or to new knowledge or advanced knowledge that is not ready to be taken by this user at his/her current level (De Bra 2002).

A learning environment can be considered adaptive if it is capable of achieving three tasks; these are: 1) monitoring the learners' activities and interpreting those activities based on specific models, 2) inferring a learner's requirements and preferences from those models, and 3) acting upon this available knowledge about the learner and the subject matter to facilitate dynamically the learning process (Paramythis and Loidl-Reisinger 2004).

Many empirical studies (Boyle and Encarnacion 1994; Eklund and Brusilovsky 1998; Brusilovsky 1998 ; Conlan and Wade 2004b) showed the added value that adaptation brings to the Web, especially in the context of e-learning. More precisely, adaptive navigational support could accelerate the user's navigation and learning while adaptive presentation can improve understanding of the content (Boyle and Encarnacion 1994). However, it is not the magical wand that solves all the Web-based e-learning problems, as stated below:

> *"creating 'good' content, a sound concept hierarchy, and the right prerequisite relationship still requires (pedagogically skilled) authors and teachers."* (De Bra 2002).

Hence, learning good practice in educational content development is still a key element for the success of AEHS.

2.4 Three Generations of AEHS and the Current Challenges

The development of adaptive educational hypermedia can be classified into three generations (Brusilovsky 2004a). The first generation systems (1990 – 1995), were pre-Web systems, whereby two main streams existed (Brusilovsky 2001a). In the first stream, researchers in the area of intelligent tutoring systems (ITS) were trying to extend student modelling and adaptation approaches developed in (ITS) with hypermedia components. In the second stream, researchers were working on educational hypermedia as an attempt to make their systems adapt to individual students. Examples of this generation's systems included: MetaDoc, KN-AHS, ITEM/IP and C-book (see in (Brusilovsky 1996)).

The majority of the next generation systems (1996 – 2002) were Web-based and more mature (as adaptive systems) than were the first generation. Examples of systems developed during this period included: ELM-ART, Interbook, PT, TANGOW, AHA!, KBS-Hyperbook, AST and Multibook (see (Brusilovsky 2004a)).

The third or current generation, starting from 2002 onwards, has been defined by the challenge of integrating adaptive hypermedia technologies into the regular educational process. Examples of this generation included: WHURLE (Brailsford 2002b) (chapter 6, section 6.2), APeLS (Conlan 2002) (section 2.9.2), ADAPT² (Brusilovsky 2005) (section 2.9.1), and MEDEA (Trella *et al.* 2005) (section 2.9.4).

Those systems demonstrated many different ways to integrate adaptation technologies but they also had many limitations, as described in (Brusilovsky 2004a). Brusilovsky claims that almost every function that is performed by a Learning Management System (LMS), such as WebCT or Blackboard[3], could be done more effectively through an adaptive Web-based hypermedia system. However, LMS is dominating the field of e-learning over adaptive hypermedia systems. The reason for this is that AEHS does not answer the demands of modern e-learning; he states:

> *"It seems obvious that the problem of modern adaptive systems is not their performance, but their ability to answer the needs of practical Web-enhanced education"* (Brusilovsky 2004a).

Some limitations of AEHS are considered responsible for their absence compared to LMS or other e-learning systems despite the great functionality, personalisation, and guidance that they provide. A number of reasons have been stated in (Brusilovsky 2004b). One reason is the lack of integration, This means that unlike the single login

[3] Both WebCT and Blackboard are now owned by Blackboard, http://www.blackboard.com/us/index.bbb

environment provided by LMS where teachers, administrators and learners use only one system that incorporates a number of ICT[4] functionalities, in AEH each system has so far focused on a specific aspect or feature of an e-learning system, requiring the use of multiple e-learning systems by a single teacher or institute.

Another problem is the limited support for the reusability of learning content as well as for the reusability of the systems' components. If a teacher is interested in the user modelling functionality of the system, this component could not be approached on its own; the system has to be used as a whole. Moreover, adding a new content fragment to an LMS is straightforward, while in AEHS it would require this fragment to be indexed with regard to the domain model concept. Even as AEHS continue to develop to overcome those limitations, full reusability as presented in courseware reusability systems and repositories, remains an issue. This limited interoperability between different adaptive systems or techniques has lead to the problem of open and closed corpus (section 2.2.4) which results in the high costs of AEHS (Brusilovsky 2007).

These aforementioned limitations indicate that the majority of AEHS were not designed for the modern e-learning context, which demands reusability of content from multiple resources. On the contrary, the adaptation is largely embedded with the content itself and the goals and objectives are pre-defined for a specific course during design time. On the other hand, reusability frameworks that have been developed to allow this open access to learning content focus on reusing static content and not on dynamic activities and changeable content such as AEH content (Brusilovsky 2007b).

[4] Information Communication Technology

The issue of interoperability and reusability is in the focus of many researchers in the AEH community. Conlan, for example stresses the importance of reusability of adaptive content given that the development of new adaptive learning content is expensive and time-consuming (Conlan and Wade 2004b). This is due to the authoring of adaptive content being complex and above the level of ordinary users, who include the majority of non-IT teachers. In some cases this authoring process requires creating multiple versions of the same lesson as in the Felder-Solomon (Felder and Soloman; Felder 1988) (visual-verbal) learning style adaptation. Hence, the authoring problem (Stewart *et al.* 2005) is another key issue that is preventing AEHS from being part of mainstream education. Therefore, researchers have tried to address this by either introducing new authoring tools such as NetCoach (Weber *et al.* 2002) or alternatively by allowing the reusability of adaptive content (Stewart *et al.* 2005). This will not remove the initial authoring costs but will limit this expensive and complex content development process to a single attempt for any given course, which is known as the 'create once and use many' approach.

In addition, AEHS are often criticised for the way they were implemented and delivered; their prototypic interfaces and experimental designs would not gain them any credit especially when compared with both open source and commercial LMS (chapter 3, section 3.3). Moreover, they lag behind the latter in providing tools for collaborative learning and keeping up with new Web 2.0 tools and technologies. This issue is addressed in greater detail in chapter 3.

The new research direction in adaptive hypermedia and e-learning is to allow the user to construct their own learning environment from a selection of services. This involves separating the content and other components, such as the user model, from the delivery service. The ability to use external content or to expose content prepared for internal use with local systems for external use is not an easy task in AEH. While the packaging of simple and static content has long been addressed by industry e-learning standards such as SCORM[5](ADL 2004), which is described in section (2.7.1) of this chapter, interoperability of adaptive learning content is more complex (Rey López *et al.* 2007) due to its dynamic and transitive nature and hence requires a lot of effort and consideration, technically and semantically. Although the focus so far was on reusing the content, which is also the focus of this thesis, nevertheless the same is valid for the users' information and hence the importance of interoperability of users' models (Zakaria 2003; Stewart *et al.* 2006). Reusability of learning materials and portability of user information are key requirements for all modern e-learning systems.

Since reusability of adaptive learning content and interoperability of AEHS components and functionality are two of the key challenges responsible for the low adoption of AEHS, greater attention is given to them throughout this thesis. The following sections analyse the different approaches used to address those issues.

2.5 Interoperability and Reusability in Adaptive Educational Hypermedia Systems

Interoperability allows adaptive systems to communicate with each other and exchange user's data, system's functionalities, and learning content. As explained in the previous section, it is one of the big challenges facing the AEH community. Interoperability in general is required at three levels: the technical level (making it possible technically), the

[5] Sharable Content Object Reference Model , from http://www.adlnet.gov

semantic level (understanding the other system), and the structural level (components and their interdependencies). Many AEH research groups are trying to tackle this issue using a variety of approaches. This thesis provides a new classification for different approaches to achieve interoperability in AEH. Four categories have been introduced which are: the Conversion or Translation approach, the *Lingua-franca* approach, the Universal Language approach and the Language-Specifier approach[6]. Each of these approaches is briefly explained below:

- The Conversion Approach: explicitly defines mappings between individual pairs of adaptive hypermedia systems. Examples could be seen in the conversion between the two adaptive systems: MOT and WHURLE (Stewart *et al.* 2004; Cristea 2005). Here the commonalities between two given systems need to be identified before the conversion occurs.
- The *Lingua Franca* Approach: this involves translating each adaptive hypermedia system to an intermediate exchange format, such as CAF (The Common Format) (Cristea 2005) and LAG (Layered Adaptive Granulation) (Stash *et al.* 2006). This then implicitly defines a translation between any two adaptive hypermedia systems, via the common format.
- The Universal Language Approach: this is an approach where differences between adaptive hypermedia systems either do not exist or every adaptive hypermedia system can directly make use of data specified in the universal

[6] This classification has been published in the International Journal of Learning Technologies as shown by the following reference:

Meccawy, M., Stewart, C., and Ashman, H. Adaptive Educational Hypermedia Interoperability and Content Creation with a Web Service-based Architecture. *International Journal of Learning Technology (IJLT)*.2007. 7 (3): p. 269 - 285.

language without further translation. This is similar in principle to a global ontology, with all applications subscribing to a common data structure and collection of operations. Although no such language exists for AEH, research on using e-learning industrial standards (Rey López *et al.* 2006) (section 2.7), or using Semantic Web technologies (Henze 2005) (section 2.8) could fall into this category since they are often working towards the same goal. However, they often require expansions and extensions to suit different requirements.

- The Language-Specifier Approach: a language specifier (for example XML) is a more flexible and lightweight approach. It prescribes how each application's language should be interpreted, to enable, for example, automatic translation of content or user's data, at least on a technical level, although often not on a semantic or interdependency level. The use of Web services that provide open APIs (such as WSDL[7]) can be seen as an example such as in MEDEA (Trella *et al.* 2005). Other research initiatives that could be classified as following the same approach provide APIs to access their adaptive systems but use proprietary (original) protocols for communications instead of agreed standards such as SOAP or WSDL. ADAPT² (Brusilovsky 2005) can been seen as an example of these research initiatives or solutions.

The last two options are a more Web-centric approach, and indeed the Web itself is based on a small number of universal languages and language-specifier such as HTML and XML. As reported earlier, there have been experiments with the conversion approach, translating MOT to WHURLE (Stewart *et al.* 2004). Moreover, an

[7] Web Service Description Language

investigation to this approach and an implementation using Web services to wrap the conversion program are presented in chapter 5 (section 5.2).

These four approaches are revisited in chapter 9 where their advantages and shortcomings are discussed further.

However most of the solutions provided by different research groups mix one or more of approach, for example the use of e-learning standards and Web services as in (Gütl 2008). Moreover, the majority of recent developments and research efforts are not only using these approaches between existing systems, but are considering rather broader solutions at an architectural level as will be described in the following section.

2.6 Architectural Solutions for Modern e-Learning Systems[8]

Although interoperability of learning content has been addressed earlier by standards as well as open repositories, the issue with AEHS in particular is that it is more than just content; it involves different components of the adaptive systems including user profiles and the adaptation rules. This interdependency problem exists because each component of an AEHS relies on the others in producing the adaptation effects. Therefore, if a course is to be reused as it is (i.e. with the adaptation effects and not just the learning materials) it is not sufficient to translate the content without the adaptation. The interdependency problem is what mostly distinguishes AEHS interoperability problems from the interoperability of other e-learning systems. The complexity of the reuse of rich, adaptive or intelligent content compared with simple content is holding adaptive and interactive systems back although those systems can be seen, from a pedagogical and learning

[8] This includes adaptive and non-adaptive e-learning systems

prospective, as having the greatest positive effect on the learners (Rey López *et al.* 2007). Hence the decision has been made by different research groups to tackle the problem at an architectural level taking into account the lessons which have been learned from the previous generations of AEHS as well as current solutions for e-learning systems in general.

One of the oldest well-known generic models for AH is AHAM (Adaptive Hypermedia Application Model) (De Bra 1999), which is a reference model for adaptive hypermedia based on Dexter's Model as described in (De Bra 1999). It took into account most existing functions supported by adaptive systems at that time in 1999. AHAM distinguishes navigational links from the content. It has a domain model, a rich user model and a teaching model. It aims to define a framework for developing adaptive systems by logically separating different design concepts. To date, AHAM remains one of the most recognised models for AEHS and new architectures and adaptive layers are still being built over this model such as in (Grew *et al.* 2008). AHAM is illustrated in figure (2.3):

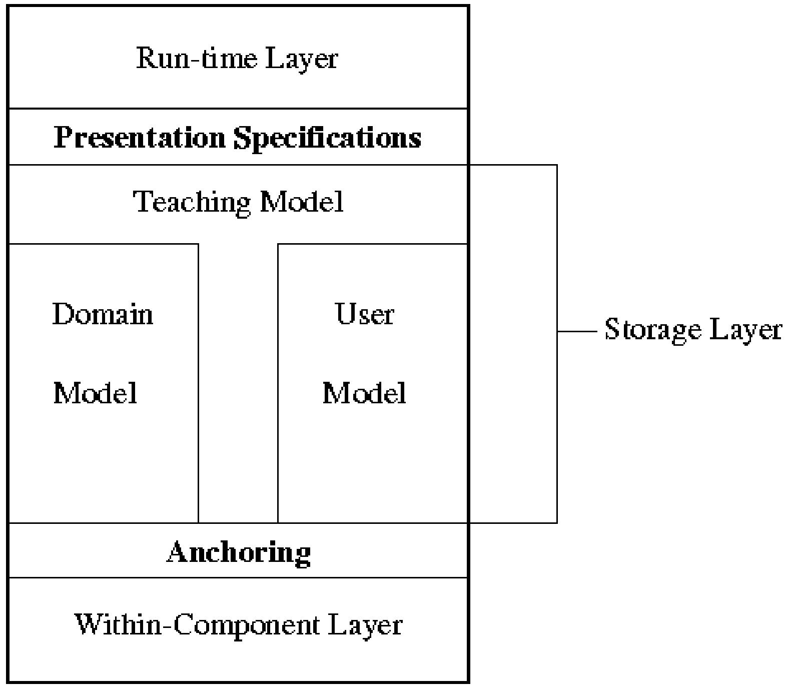

Figure 2.3: *"The AHAM Model"*, taken from (De Bra 1999).

The LAOS (Cristea and De Mooij 2003) layers model has been built on the former AHAM model and has gone into a greater depth in terms of concepts separation. It consists of five layers which are: the Domain Model (DM), the Goal and Constraint Model (GM), the User Model (UM), the Adaptation Model (AM) and the Presentation Model (PM). LAOS was generic since it captured the main conceptual components constructing an adaptive system. However, alongside the other existing AEHS, the LAOS implementation by Cristea in MOT (Cristea and De Mooij. 2003), which implements a simplified three-layer model of LAOS, separated those aspects of the system conceptually. LAOS is presented in figure (2.4):

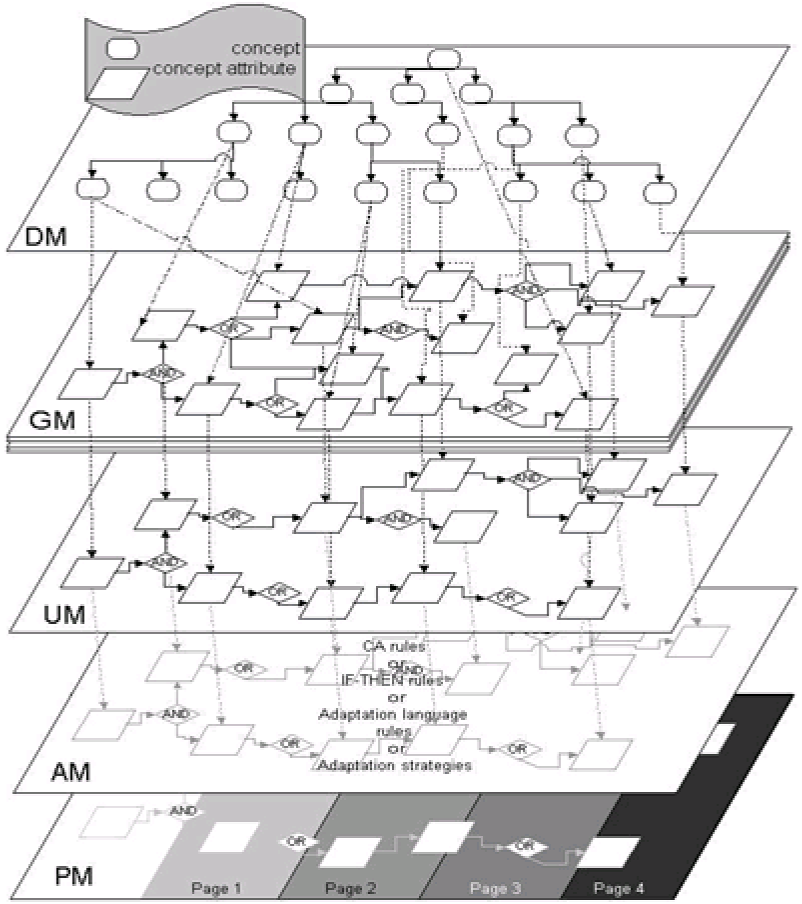

Figure 2.4: The LAOS Model, taken from (Cristea and De Mooij 2003).

A step further from the LAOS model, modern AEHS separates systems' components conceptually and physically and there is a new focus on component-based architectures. This approach is seen as a possible way for AEHS to penetrate into classrooms. Distributed component-based architectures for AEH are defined in order to assemble courses dynamically during run-time (Brusilovsky 2003a). Hence systems such as KnowledgeTree (Brusilovsky 2004b) which evolved later into ADAPT² (Brusilovsky 2005), APeLS (Conlan 2002; Conlan *et al.* 2002), Personalization Services (Dolog *et al.* 2004; Henze 2005) and MEDEA (Trella *et al.* 2005) have been introduced. A short survey of those systems is presented in section 2.9 of this chapter.

The problem here is to develop an architecture that would allow independent developers or research groups to create adaptive courses that could be accessed by different systems and work in parallel with the same user while exchanging information about this user, resulting in more efficient and effective adaptation (Brusilovsky 2005).

As argued in (Conlan *et al.* 2002), AH systems can be viewed as personalised content services with the capacity to deliver content to/from many learning environments and not only to serve content from their local repositories through their local interfaces. Providing personalised learning using distributed information in a dynamic and heterogeneous environment has a number of different requirements compared to traditional monolithic systems which have preceded them and is considered more complex. Service Oriented Architectures or *SOA* (chapter 4, section 4.4) became a recommended approach to deal with this problem in e-learning systems (Millard *et al.* 2008). Moreover, this type of architecture has often been used with other technologies, standards or specifications, as will be described shortly.

In SOA, a distributed system is built as a composition of interacting services. Well-defined interfaces provide a mechanism to access functionality and define contacts between services. Those interfaces are usually built to be platform-independent and can be dynamically located and invoked. Moreover, they have the ability to deal with functional aspects of the systems in terms of run-time environment, registry, transport and communication protocols (Gütl 2008). Therefore, SOA-based systems have the advantages of being loosely-coupled, distributed and sharing a common set of interfaces standards or specification. These loosely-coupled learning services provide flexibility in terms of integration. In addition, they allow for modularity, interoperability and extensibility (Wilson *et al* 2004 as cited in (Millard *et al.* 2008). This has even been supported by JISC[9], which is developing a common service framework for e-learning, e-research and e-administration known as the e-Framework (Oliver and Hawker 2005). This framework and others aims to be populated with services by the e-learning community itself. Hence this makes the reusability of these services possible and lead to much easier developments of new learning courses. Web services in particular, as one of the most common implementations of SOA, have the technical ability to make this process achievable and less complex. They are useful tools for defining e-learning frameworks because of their open interfaces, modularity, simplicity and flexibility.

New architectures have been defined to find solutions to the current limitations faced by different e-learning systems. Although not specifically designed to cater for adaptive learning in particular, which is a special case of online learning, some of those solutions can be adopted by the AEH community, fully or partially, at least to address common problems that are shared by all e-learning systems. Hence some of those solutions will be reported here. For example, the research in (Kayama and Nagata 2008) analysed many e-learning platforms, architectures and systems. The authors claim that such architectures have their own limitations in terms of scalability, availability and distribution of resources, and hence propose a system that follows *Grid* architectures as an alternative. In Grid architectures, individual users and applications access resources with little or no knowledge of where the resources are located or what is the underlying platform. The new system is known as GLORIS, which combines a SOA with WSRF[10] and Web Services.

---

[9] Joint Information Systems Committee *(*JISC*),* from: http://www.jisc.ac.uk/
[10] Web Service Framework, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf

Moreover, another research used pattern languages for e-learning systems (Harrer and Martens 2008). It divides solutions for the interoperability problem into two categories: standardised solutions (IEEE LTSC[11], or Dublin Core Metadata Initiative DCMI[12]) and Software Engineering (SE) methods. The first approach has the advantage of a pre-definition of terms, which is a step towards exchangeability. However, the contributions remain at the level of those terms and lack a specific description of how to realise and implement the systems parts. On the other hand, in SE the development of software is based on techniques which are clear, definite, and traceable. Hence the research in (Harrer and Martens 2008) considers this approach as more appropriate for defining general, workable, and interoperable e-learning system by defining those language patterns. It is supported by work conducted in (Laforcade *et al.* 2008), which provides a mode-driven engineering and a mode-driven architecture (MDE-MDA) for technology-enhanced learning. In addition, this view is shared in (Oubahssi and Grandbastien 2008) which states that SE provides models, methods and tools for implementing new software and by re-engineering those models, new solutions will evolve from existing ones, which is more efficient and useful than creating new models. Moreover, it reported two main problems faced by the e-learning community: lack of open systems due to the closed vision of some e-learning services, and lack of consensus at an international level in terms of standards despite the existence of some commonly agreed specifications such as AICC, IMS or SCORM. Since these specifications or standards do not cover or address the whole e-learning delivery process. Specific services and functionalities are yet to be defined to allow interoperability between e-learning systems such as a Content

[11] Leaning Technology Standard Committee, from: http:// www.ieeeltsc.org
[12] http://dublincore.org

Management Service (CMS). The CMS relates to the mechanism of content declaration at the level of a provider system and the mechanism of external access to those contents. The new architecture that is described by this thesis also defined a CMS as will be seen in chapter 6.

Although from a technical point of view, the idea of having open and distributed systems seems promising and achievable, it might raise other concerns in areas such as legal issues. Legal issues can be raised concerning the idea of interoperable user models, which means that some personal information about a specific user is collected explicitly or implicitly and can then be accessed or shared by different systems with different levels of security (De Bra 2004).

For e-learning platforms to be future-proof, they should be able to support a wide range of needs by providing interoperability architectures for the various existing and emergent e-learning services. This should be done on different levels (intra-domain and inter-domain) provided by the separation of concepts and functionality such as authoring and delivery (example LMS and LCMS) (Dagger *et al.* 2007). These services apply a number of technologies and techniques, have a wide range of teaching and objectives and goals, and are accessed using different devices and interfaces. These include adaptation, personalisation, mobility, and simulations.

As concluded by Gütl in (Gütl 2008), the separation of logical and technical systems seems promising in terms of openness, flexibility, extensibility, inter-changeability and scalability. Despite the additional efforts required in developing SOA for AEHS in terms of managing and invoking services, they provide a good solution for openness and flexibility in e-learning environments. Moreover, the application of service-based approaches is seen as a cornerstone in the movement towards a generic e-learning system for flexible online teaching and learning activities. Services allow for easy selection, combination and replacement of specific functionalities and capabilities of such services. This results in the creation of specialised e-learning applications and solutions. However, issues such as privacy, security and reliability might require further enhancement to those architectures.

The previous sections looked at ways in which interoperability in the context of adaptive learning has been addressed with a greater focus on redesigning the architectures to make them open and flexible. Those architectures used a number of technologies in order to achieve this goal, such as Web services (as seen in section 2.6), e-learning standards and Semantic Web. Both the e- learning standards and the Semantic Web will be explained in the following sections 2.7 and 2.8 respectively.

2.7 E-learning Standards and Adaptation

In previous sections, e-learning standards have been mentioned as a possible solution to e-learning interoperability problems. There has been a wide adoption of some of the standards such as ADL SCORM, which is used by both modern adaptive and non-adaptive e-learning systems, either on its own or by mixing it with another solution. This strong presence of e-learning standards in general within the new architectural solutions made it worth explaining further.

A number of standards have been defined by IEEE LTSC (Learning Technologies Standardization Committee)[13], IMS Global Learning Consortium[14] AICC (Aviation Industry CBT Committee)[15] or ADL (Advanced Distributed Learning)[16]. SCORM is the most popular standard, it has been used in a number of researches such as in (Conlan 2002; Rey López *et al.* 2006; Gütl 2008), and hence it is going to be focused on here.

2.7.1 SCORM: An Overview

SCORM™ stands for Sharable Content Object Reference Model, and it is a standard for Web-based e-learning content sharing. It defines how the individual instruction elements are combined on a technical level and sets conditions for the software needed for using the content.

SCORM is developed by the Advanced Distributed Learning Network (ADL),which defines SCORM today as:

> *"...a collection of specifications and standards that can be viewed as separate "books" gathered together into a growing library. Nearly all of the specifications and guidelines are taken from other organizations. These technical "books" are presently grouped under three main topics: "Content Aggregation Model (CAM)," "Run-Time Environment (RTE)" and "Sequencing and Navigation (SN) (introduced in SCORM 2004)" "*.. (ADL 2004)

There are a large number of LMS that support SCORM. Therefore, SCORM-compliant learning content can be delivered to many different LMS without modification. This

[13] http://ieeeltsc.org/
[14] http://www.imsproject.org/
[15] http://www.aicc.org/
[16] http://www.adlnet.gov/

provides greater interoperability between different LMS and thus allows for the reusability of learning content. A SCORM-compliant course should satisfy the four main characteristics of SCORM known as RAID: Reusable, Accessible, Interoperable and Durable (2003)

To describe a course that follows the SCORM standard requires reconstructing the course from the beginning, which can be done either at a low level or at a high level. On a low level the process is done by finding a mapping of existing resources to SCORM content model components, writing appropriate XML documents and adding physical files to construct SCORM Content Packages (Arapi *et al.* 2003). On higher levels, this can be done using any of the available tools for constructing such courseware, for example Reload Editor[17]. To conclude, SCORM facilitates the development, packaging and delivering of Web-based learning content when and where they are needed.

2.7.2 A Brief Overview of SCORM characteristics

SCORM is unique for some of its design and functional characteristics. Here is a list of some of the most important of them (LP and CANAL 2003):

- A central design characteristic of the SCORM specification is the division of responsibilities between the LMS and the SCORM compliant content.
- SCORM defines a standard communication mechanism between the LMS and the content, which makes it feasible to re-use content with different LMS.

[17] http://www.reload.ac.uk

- Within a SCORM compliant course, the SCO is the smallest reusable learning object that can communicate with LMS in order to perform data exchange or data tracking.
- Elements in SCORM's data model are based on the AICC specification and are restricted to each individual SCO and each individual learner. Two different learners cannot share any of these data elements to pass information between themselves. SCORM assume a situation with a solo learner and does not by itself support collaborative learning.

Figure (2.5) illustrates SCORM architecture while figure (2.6) presents a diagram about the process of launching SCORM-compliant content by an LMS.

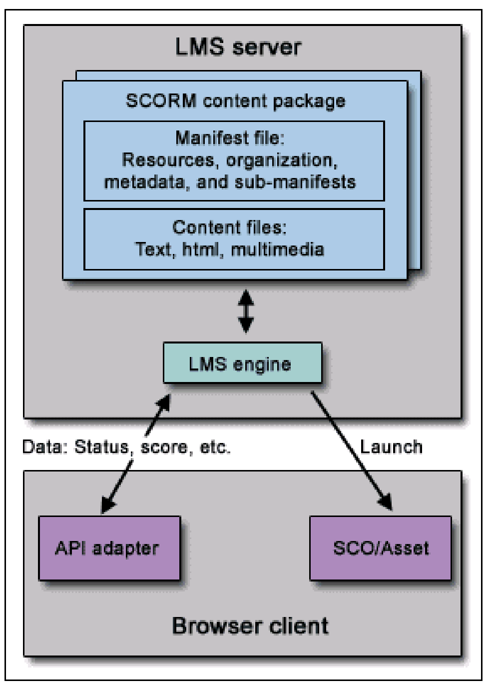

Figure 2.5: SCORM, taken from (TechRepublic 2002)

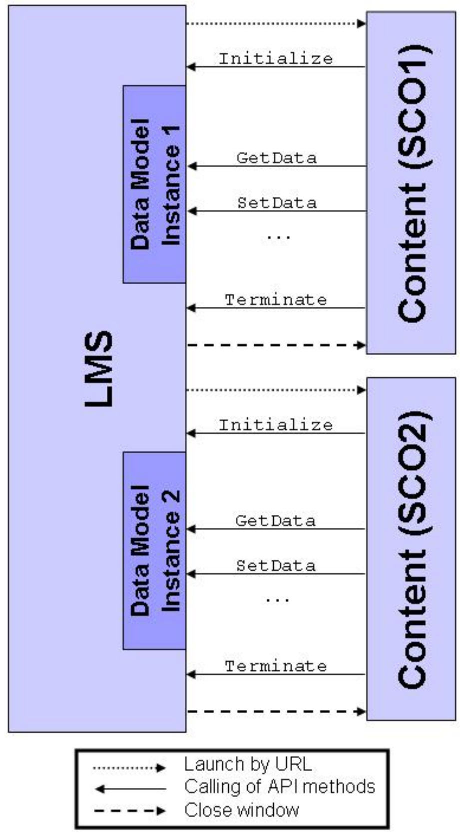

Figure 2.6: Launching SCORM-compliant content by a LMS , taken from (Rey López *et al.* 2007).

2.7.3 SCORM, AEH and Interoperability

One of the earliest research on the use of SCORM in AEH has been done by comparing the possible adaptation achieved by AH and also by SCORM 1.3 which has adopted IMS Simple sequencing in (Abdullah and Davis 2003). While AH systems employ a user model and a model of concepts in the learning domain to decide what navigational features to display and how to present the content, SCORM provides a learner with a sequence of learning activities to guide him/her through a learning space to achieve a specific learning object. This sequencing is based on instructional design strategies expressed in the manifest XML file. The LMS then maintains a simple user model based on learning objects that the user has visited as well as the intermediate test results; this will then be used to decide the following object. This comparison concluded that although both approaches might look similar on the surface, they are fundamentally different. Those differences have been explored from different angles: from the objective of the approach, components, conceptual structure, techniques employed, tracking mechanisms and reusability. In general the sequencing falls short of the objectives of the AEH community to provide personalised learning since the user model is simple and does not provide persistence between objects. Moreover, SCORM confines learners to a static pool of learning resources organised in a pre-defined narrative structure, resulting in limiting the user in exploring other topics around the current one. Hence further research has conducted in (Abdullah *et al.* 2004; Abdullah and Davis 2005) to augment SCORM with AH and to add links dynamically to the SCORM manifest to compliment its pre-existing content. This new architecture provides SCORM with a service that supplies those links in run-time and introduces the learner to alternative resources. This service is known as Personalization Link Service (PLS) (Abdullah and Davis 2005).

More research on SCORM and adaptation can be found in (Rey-López *et al.* 2008), where the advantages of SCORM as a widely-adopted e-learning standard is recognised and highlighted in addition to its limitations. This research provides an extension to SCORM to create adaptive courses and SCOs for adaptive courses; it presents a set of options with adaptation rules to decide which is most appropriate for the student. For adaptive SCOs, it extends SCORM-RTE data model to allow SCOs to access the user's characteristics and change its behaviour accordingly. It states a number of advantages over formal proposals because of its encapsulation of the adaptive information and its modularity for both individuals and systems.

There is a debate within the AEH society regarding the use of standard protocols to achieve this interoperability between systems. Brusilovsky argues that the standards-based model is not suitable for adaptive distributed content (Brusilovsky 2004b), while on the other hand, others (Conlan 2003; De Bra 2004) support the use of open standards and protocols.

SCORM's limitations could be generalised to some extent to include the rest of the standards. When these standards are applied in e-learning systems they achieve high performance when dealing with simple learning objects such as a text file, picture or a simple animation. On the contrary, when dealing with more complicated content such as rich intelligent or adaptive content, they show many limitations because of the dynamic and changeable nature of this type of content. Research described in (Rey López *et al.* 2007) presents ADL SCORM and IMS SSP (Sharable State Persistence[18]) as two examples of e-learning standards investigated in the road to adaptive learning in the context of RTE, and it compares their limitations with the flexibility and feasibility of non-standard research solutions. In (Dagger *et al.* 2007), the authors have criticised standards for missing a crucial component in terms of interoperability, which is the ability to use the data and information dynamically in a meaningful way (semantically); then they continue to promote the contribution of the Semantic Web to fill this gap. Semantic Web is addressed in the next section.

[18] http://www.imsglobal.org/ssp/index.html

2.8 The *Semantic Web* role for interoperability in e-Learning

Since the use of Semantic Web is becoming more popular as a new research direction in AEH to address the issues of interoperability and semantic-exchange, this section is dedicated to the application of Semantic Web in AEHS. The Semantic Web (Berners-Lee *et al.* 2001) is, according to the W3C, a Web of data which provides a common framework that allows this data to be shared and reused across applications, enterprises, and communities. It adds:

> *"The Semantic Web is an evolving collection of knowledge, built to allow anyone on the Internet to add what they know and find answers to their questions. Information on the Semantic Web, rather than being in natural language text, is maintained in a structured form which is fairly easy for both computers and people to work with."* (W3C 2002a).

The Semantic Web can be understood as a vision where formally-described data on the Web can be readable and comprehensible by the computer as a machine rather than by humans. The machine is able to consume such data and deduce or infer relationships, meanings, reasons and answers to proposed questions, automatically without human interferences. Since the focus is on the data and not the processes, new technologies and protocols are introduced to reach a common understanding of this shared data. These include but are not limited to RDF[19] and OWL[20].

A distinction between Semantic "Web Services" and "Semantic Web" services is given in (Millard *et al.* 2008), where the former refers to Web services that apply the Semantic

[19] Resource Description Framework, from: http://www.w3.org/RDF/
[20] Web Ontology Language, from: http://www.w3.org/TR/owl-features/

Web technologies and the latter are services designed to use the information on the Semantic Web.

In the context of e-learning, the Semantic Web is used as a reference for e-learning semantics. It aims to allow collaboration and interoperability through semantic meta-data applied to the learning resources. The Semantic Web here is usually applied within a SOA. A common description is needed to share this data across multiple applications using ontology. Ontology is a description or specification of a given domain and what is permitted in this specific domain. This includes concepts as well as relationships in this particular domain. Advantages of the application of Semantic Web technologies include: metadata, interoperability and reasoning (Millard *et al.* 2008). However, interoperability (issue of concern) is often quoted but rarely demonstrated (Millard *et al.* 2008). Yet it is seen as one of the possible advantages that is still pushing Semantic Web and its applications forward.

A number of e-learning frameworks applied Semantic Web technologies such APeLS (Conlan 2002) (section 2.9.2) or Personalisation Services (Henze 2005) (section 2.9.3), which are both adaptive systems. Other general (non-adaptive) e-learning systems include Question Bank (Millard *et al.* 2008) which provides a combination of Web services and Semantic Web technologies in a SOA, and argues that "Semantic Web" service has a lot to offer in terms of e-learning: easy development and easier mark-up resources to allow for external access and manipulation of the learning services.

The main link between AEH and the Semantic Web is the Web services' technologies (Dagger *et al.* 2007); they allow for remote systems to communicate with adaptive services to create complex systems built from atomic services. Those adaptive services take the advantages of the service ontology provided by OWL-S. It supplies them with a rich meta-data description and highly expressive framework for interoperability at a semantic level as well as at a technical level. This is applicable in terms of user's data as well as learning content materials and links within a learning system.

On the other hand, a number of researchers are doubtful about the real possibility of agreeing on a common ontology (Marshall and Shipman 2003) and hence having a true Semantic Web working in practice. There are many issues surrounding the Semantic Web, critics question the practical feasibility of a partial fulfilment of what it promotes. Hence, consider it as an unrealistic idea and some critics even argue that there a real need for a Semantic Web. However, others who might believe that the Semantic Web is capable of delivering what it promises have problems with issues such as censorship and privacy. They believe that machine readability would make it much easier for information to be understood by automated content-blocking programs, hence can be used by governments to control Web access. Moreover, the geo-location meta-data can reveal the user's location and therefore violets his/her privacy. Other developers, who might not be concerned with these issues but worry that the Semantic Web would make it much more time-consuming to create and publish content on the Web because of the need for two formats for humans and machines.

Nevertheless, although a lot of debate surrounds the Semantic Web in general, there might be more possibility of it becoming a reality in specific controlled domains, such as in a specific organisation (intra-company project) if not in the Web as a whole.

As mentioned earlier, the Semantic Web and the standard approaches are both working towards a goal of having a Universal Language (section 2.5). However, another problem of a common language is that it could expand and become unreasonably large as will be described further in chapter 9.

2.9 Modern SOA Adaptive E-learning Frameworks: A Brief Survey

A number of adaptive learning SOA systems or frameworks started to emerge at the beginning of the third generation of AEHS and new systems continued to be defined, developed or enhanced to the current date (2008). Below is a short description of some of those systems frameworks.

2.9.1 ADAPT²

ADAPT² stands for Advanced Distributed Architecture for Personalized Teaching and Training (Brusilovsky 2005). It is a distributed architecture for adaptive e-learning based on reusable intelligent activities. Former versions were known as KnowledgeTree (Brusilovsky 2004b). ADAPT² has been created to accommodate a larger variety of adaptive components by applying a higher level mechanism for ontology-based interoperability between self-contained adaptive Web-based systems. It uses proprietary (original) protocols for communications between different components. Since ADAPT² is one of the main players in the case study presented in chapter 5, this system is explained in greater detail (section.5.3.2).

2.9.2 APeLS

APeLS: Adaptive Personalized e-learning Services (Conlan 2002) is developed as a service to deliver personalised educational courses based on a multi-model, metadata-driven approach that makes use of some e-learning standards as well as some of the Semantic Web technologies. With the separation of content, portal and narrative (or expertise: this refers to concepts that might be selected as a part of the course), it facilitates the maximum reuse of the narrative and content. Since APeLS was used in the case study presented in chapter 5 (section 5.3), this system is explained in greater detail in that chapter (section.5.3.2).

2.9.3 Personalization Services

Personalization Services as presented in (Dolog *et al.* 2004) is an adaptive educational system describes an approach that aims to fill the gap between current adaptive educational systems and open, dynamic learning repository networks. This approach is based on Semantic Web technologies and it also extends a number of e-learning standards. The main focus has been on interoperability issues for discovering and integrating learning resources (services). The overall architecture, presented in the following figure (2.7), applies a decentralised approach for integrating the system's components and the distributed learning resources. Regarding the semantic description of the architectural components, this approach describes the learning resource, the knowledge domain and the learner's characteristics using the Semantic Web technologies: OWL[21], DAML-S[22] and RDF[23]. This system is an approach to bring personalisation to the Semantic Web for the area of e-learning. It shows how functionalities of personalised learning can be embedded into Semantic Web services while supported by services for retrieving learning resources or user information.

---

[21] Web Ontology Language, from: http://www.w3.org/TR/owl-guide/
[22] DAML Services (Ontology), from: http://www.daml.org/services/
[23] Resource Description Framework, from: http://www.w3.org/RDF/

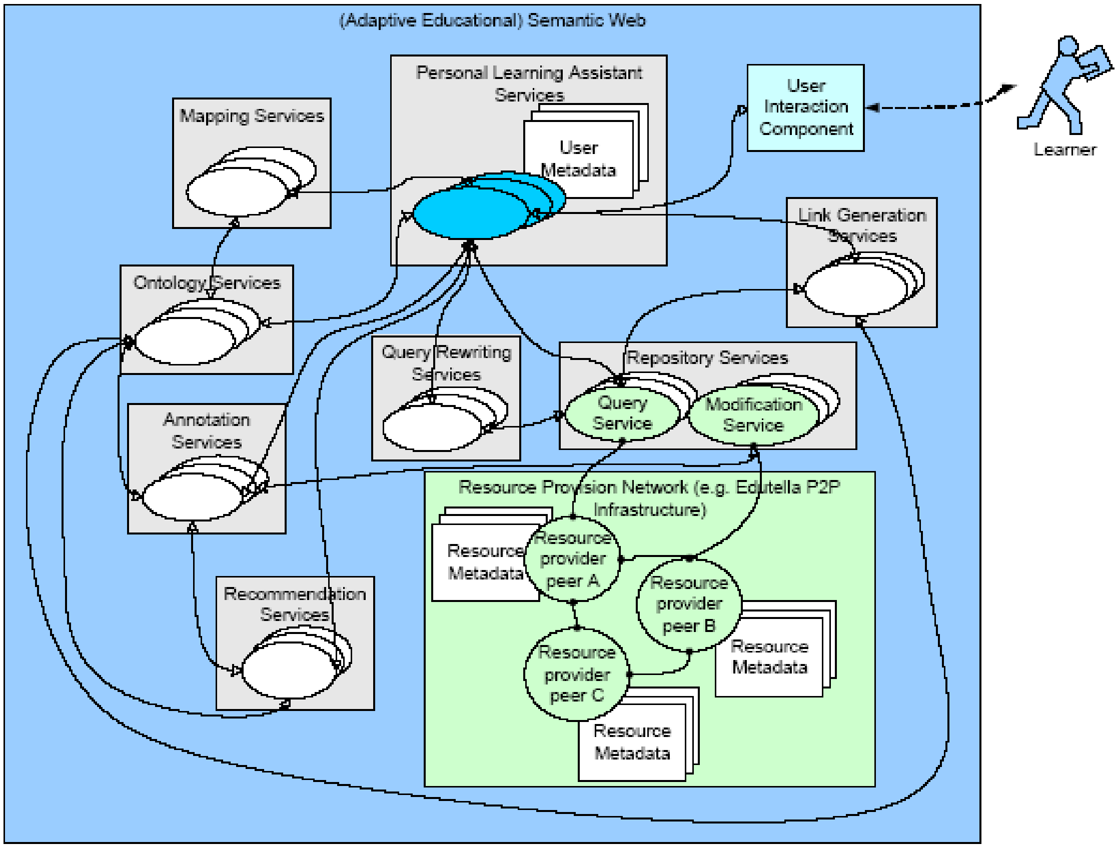

Figure 2.7: *An architecture for personalization services*, taken from (Dolog *et al.* 2004).

2.9.4 MEDEA

MEDEA is a Spanish acronym for Methodology and Tools for the Development of Intelligent Environments of Teaching and Learning (Trella *et al.* 2005), which is an open, service-based, general-purpose learning platform for the deployment of intelligent and adaptive Web-based educational systems. It does not deliver learning content which was developed locally by the same system; rather it is a service-based learning platform that allows the use of different integrated learning systems for educational and instructional purposes. It connects and communicates those systems using the open Web services protocols which allow for more interoperability, reusability and flexibility of structure. The following figure (2.8), describes the MEDEA's architecture:

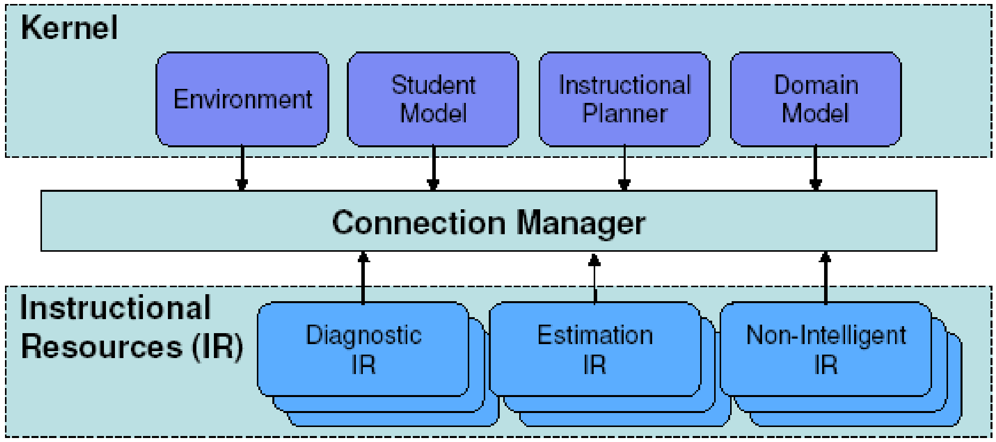

Figure 2.8: *Architecture of MEDEA*, taken from (Rey López *et al.* 2007)

The *Domain model* is the component where knowledge about the subject is stored. It is defined by a semantic network of concepts and the relations between them. For a resource to be available for MEDEA's users, it requires three main features: an overlay student model, a set of semantic concepts in the domain model and a set of services implemented as Web services and described in WSDL files. The *Connection Manager* manages all the communications between MEDEA's services. These communications can be viewed on two different levels: a software implementation level (developer's level), solved using the emerging Web services technologies, and a concept semantic level (teacher's level). MEDEA requires that the platform shows a unique view to the student, where the integration of those different resources is transparent to him/her. In addition, there should be a unified student model which collects and summarises all the data from the independent student models which belongs to each resource. Therefore the semantic relationships between the concepts of MEDEA and those of resources must be established. MEDEA provides an interface to perform this task and delegate this responsibility to the teacher who is able to deal with the semantic issues.

2.9.5 AdeLE

AdeLE**:** Adaptive e-Learning with Eye-Tracking (Gütl 2008), follows a student-centred approach to provide personalised knowledge content from static and dynamic repositories. This is achieved by using fine-grained information about its students, which includes an eye tracker system (ETS) for extending the observation while the students use the system. It applies a strong separation between the client-side (Web client, ETS and Content Tracking System or CTS) and server-side (User Modelling System or UMS, Adaptation System or AS, LMS, Information Retrieval System or IRS, Concept Modelling System or CMS) of its architecture. Moreover, existing LMS can be integrated into the AdeLE environment using the given interface definitions. An overview of the ADELE architecture is provided in the following figure (2.9):

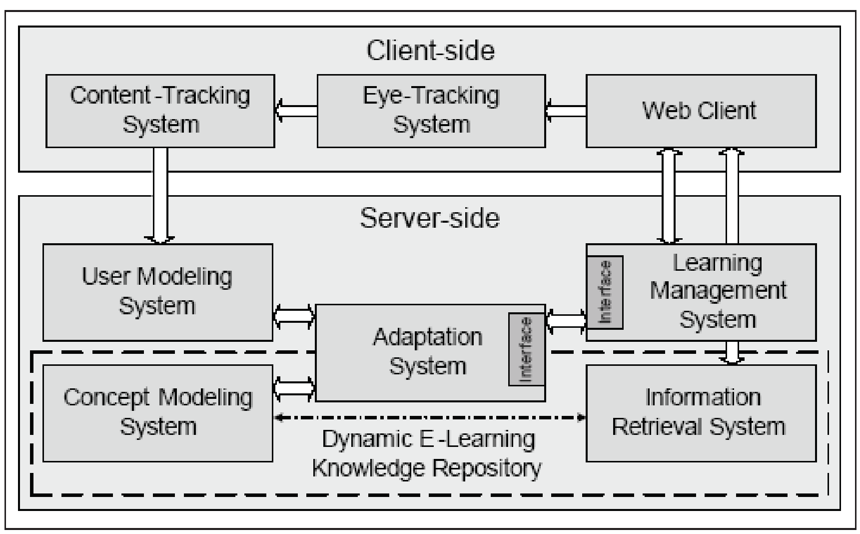

Figure 2.9: *Overview of the AdeLE Architecture*, taken from (Gütl 2008)

In AdeLE, adaptation tasks are performed according to the course model derived from the SCORM manifest file and the UMS. The adaptation functionality has been packaged using SCORM Content packaging and SCORM RTE with a set of adapters which were required for the packaging of those standards to occur.

2.9.6 DIOGENE

DIOGENE (Sangineto 2008) is built as an adaptive e-learning platform for the generation of personalised courses. With DIOGENE, the LMS automatically generates personalised courses by assembling materials using both static and statistics of knowledge. Statistical knowledge is information about the learner during the interaction with the activities which are executed at the end of each learning session. In addition, the static information about specific domains is described using an ontology through the abstract representation of a specific domain. Moreover, it uses e-learning standards through the description of each single learning object by means of metadata. The adaptation provided here follows the learner styles according to Felder-Solomon (Felder and Solomon) approach. DIOGENE's architecture is illustrated in figure 2.10 below:

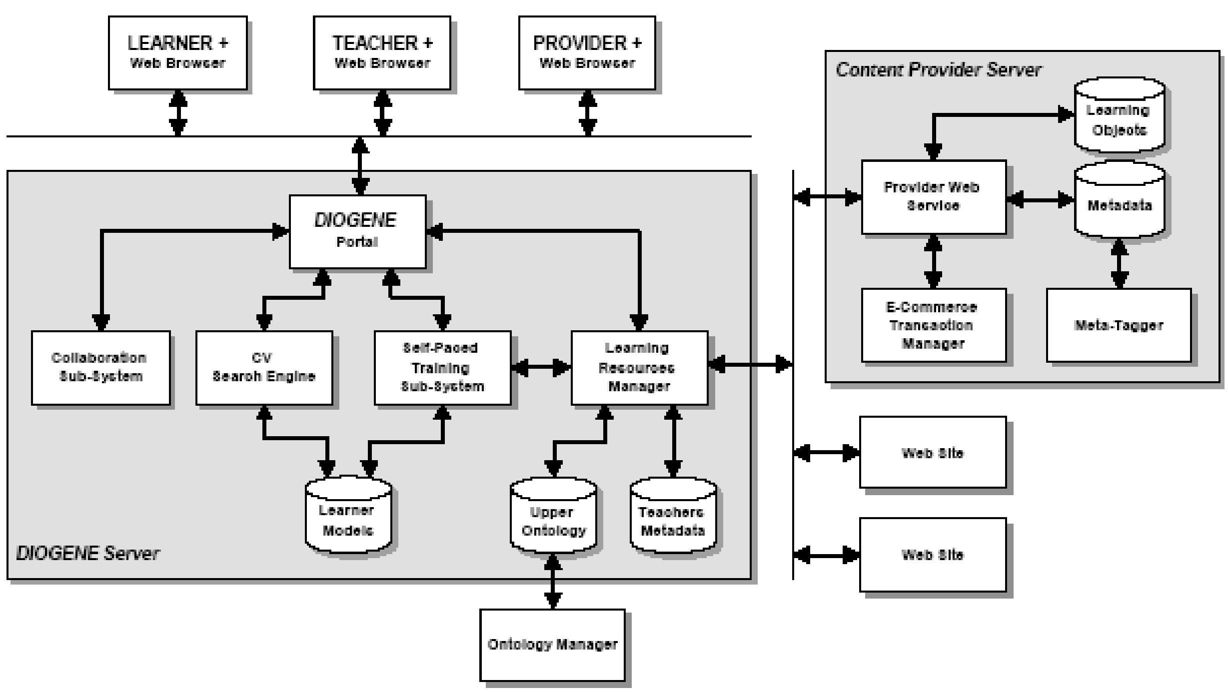

Figure 2.10: Architecture of DIOGENE, taken from (Vergara Ede and Capuano 2003)

2.10 Summary

Adaptive Educational Hypermedia is an application of general adaptive hypermedia's concepts and technologies in the field of Web-based learning. A number of studies pointed out its importance for enhancing online learning and aiding the students' ability to learn by providing them with a "tailored" version of their courses or lessons. Despite the aforementioned advantages over other e-learning hypermedia systems such as Learning Management Systems (LMS), adaptive systems have poor presence in the online learning community. The chapter looked at a number of reasons, described in the literature, and among them were AEHS complexity of authoring and the limitations in terms of reusability and interoperability at different levels. A number of research approaches have been followed in order to address the interoperability issue.

One of those approaches has involved redesigning the architectures of adaptive systems, making them more open, generic and interoperable. Re-engineering the architecture of e-learning systems in general and adaptive systems in particular is considered a valid approach to address e-learning challenges and provide sustainable solutions for real world challenges. Within these architectures, a number of technologies, standards, languages or specifications have been used such as SCORM, Web services and Semantic Web.

Future requirements for e-learning systems include a number of possibilities at all levels. The focus is predicted to be on providing adaptation according to various devices and interfaces such as mobile phones, PDA's, wireless devices and integration with media systems. In addition, adaptive systems would have multi-model approaches for users and content. Moreover, security and/or privacy of learning content will attract more research attention in the context of e-learning, pushed by legal issues. Some predict the influence of the Semantic Web will grow. Furthermore, Web 2.0 and its technologies, as well as concepts that promote sharing, collaboration and communication, would have a large impact on those modern e-learning systems.

Web 2.0 and e-learning 2.0 and their links with adaptive learning is the topic of the following chapter.

CHAPTER 3

"And help you one another in righteousness and piety"

The Holy Quran, Surah Al-Maidah 5:2

PERSONALISATION, COLLABORATION AND E-LEARNING 2.0

3.1 Overview

While the1990s introduced the Web to the world, the 2000s are introducing "Web 2.0" with its huge diversity of users who are becoming more digitally-literate, alongside the new emerging technologies in terms of hardware and software. The term Web 2.0 was introduced by Tim O'Reilly (O'Reilly 2005), and refers to an idea rather than a concrete technology. This idea comprises user-generated content, which accumulates and aggregates the power of the community or crowd (Safran *et al.* 2007). However, the term Web 2.0 remains a debatable subject in the scientific community although it has been widely adopted to refer to the new trends and technologies, which changed the user from being primarily passive to being the key player in shaping this new environment.

This new generation of the Web is focusing on people rather than applications and is known as the *socialisation* of the Web (Fumero 2006). Content is created by the users and not just the system providers, and this content is also evaluated by those users. Web 2.0 is an umbrella term, which describes new technologies that empower websites to be more than static pages; the user contributes to and interacts with the data (Chan 2007).

Hence the Web is becoming more alive by the interactivity of its users. Humans, not the software, are adding significance to the Web and its dynamic content by contributing and collaborating with one another across the globe through the Internet in a social manner. This recognition of the human's importance made the *Time Magazine* honour "You", citizens of the new digital democracy, as the Person of the Year in 2006 (Grossman 2006).

Nowadays, students access Learning Management Systems (LMS), which are equipped with some of the Web 2.0 tools and which promote learning by sharing and interacting. Adding to those rich LMS the ability to connect to external services that provide personalised and adaptive learning is seen as a valuable enhancement to those systems, which this research has set out to achieve. Hence the LMS itself becomes this virtual space where the learning and socialising takes place; it is the user's interface and delivery service of the architecture that has been introduced by this thesis.

Since both LMS Moodle (Moodle 2008) (section 3.2.1) and ATutor (ATutor 2008) (section 3.2.2) are used in implementation and testing of the new architecture, attention is given in this chapter to both systems as examples of popular open source LMS. Moreover, other research proposals, which try to integrate LMS with external systems/services, are also reviewed here.

Finally, a relatively new type of adaptation is group adaptation, which has been placed in this chapter due to its connection to collaborative learning

3.2 Web 2.0 and E-learning 2.0

O'Reilly (O'Reilly 2006) tries to provide a compact definition to Web 2.0 software:

"*Web 2.0 applications are those that make the most of the intrinsic advantages of that platform: delivering software as a continually-updated service that gets better the more people use it, consuming and remixing data from multiple sources, including individual users, while providing their own data and services in a form that allows remixing by others, creating network effects through an 'architecture of participation' and going beyond the page metaphor of Web 1.0 to deliver rich user experiences*".

A blog is a well-known Web 2.0 application which is an interactive journal-like application that has been replacing former personal Websites. Some of the other Web 2.0 services are sharing photos and videos using applications such as *flickr*[24] (photos) and *YouTube*[25](videos), building social networks of friends using applications such as *facebook*[26] and collaborative building of knowledge databases or encyclopaedias such as *wikipedia*[27]. In addition, there is the collaborative evaluation through tagging and commenting of products, information or Websites, such as Amazon.com book reviews, *del.icio.us*[28] portable bookmark lists, or question services through *Yahoo Answers*[29], as well as labelling and classification of Web content in general such as in (Truran *et al.* 2005). Moreover, with Web 2.0, the importance of personalisation is evident; users now

[24] http://www.flickr.com/

[25] http://www.youtube.com/

[26] http://www.facebook.com/

[27] http://wikipedia.org/

[28] http://del.icio.us/

[29] http://answers.yahoo.com/

expect to be automatically notified about new products and services that meet their interest from their favourite websites. In addition, they can expect their behaviour to be monitored in order to adapt to the new change of interest or behaviour, hence the growing importance of user modelling and adaptation.

Web 2.0 is changing how the world interacts, socialises and learns. Therefore, the same trends and advances that have started to dominate the Web have also affected the e-learning systems, which are becoming "learner-centred" (Downes 2005). It is moving from the *know-what* and *know-how* to *know-where* (Corcoles *et al.* 2007), away from the traditional learning experience towards training the users to gain the ability to locate information and knowledge and be able to make use of this knowledge. This type of learning aims towards a new generation of online learners, who are described as *digital-natives* (Corcoles *et al.* 2007), since they have been introduced to the Web and its technologies earlier in their education than the previous generations.

When the Web 2.0 concepts were applied to the e-learning systems, it resulted in the term *e-learning 2.0* being coined by Downes, who explains it thus:

> " *Learning is characterized not only by greater autonomy for the learner, but also a greater emphasis on active learning, with creation, communication and participation playing key roles*" (Downes 2005).

Around 95% of UK Higher Education institutes reported using at least one *Virtual Learning Environment* (VLE) (Walczowski and Ellis. 2007), which is another name for the LMS. The modern VLE or LMS provides the tools and environment where this social collaboration can take place within a learning context and therefore is essential to today's online learning. Moodle, which is chosen in this research as the LMS and login point to the new adaptive integrated learning environment, has a number of e-learning 2.0 features (Millard and Essex 2007). It provides a collection of collaborative and social tools that Moodle calls *activities*, such as Wikis, Chat, Blogs, Forums, Workshops, Glossaries and other tools. According to Alexander (Alexander 2006), education is like a conversation and learning content is something with which one performs some type of operation, rather than the learner just reading the content passively .

The distinct characteristics of e-learning 2.0 could be summarised as follows: one which is learner-centred, aimed towards digitally-literate learners, where all users communicate as peers in this social environment, and in which social software is built upon a Web platform using new Web technologies, such as Web Services and AJAX. In addition, it moves from the traditional monolithic standalone architectural learning applications with static, predefined learning content into an open learning environment of interoperable open source platforms and tools that support online social networks and communities (Fumero 2006).

Application of usage of Web 2.0 tools in different universities and learning institutes is presented in (Adisen and Barker 2007) and a usage survey to measure the impact of Web 2.0 is reported in (Safran *et al.* 2007). This survey has shown that blogs and wikis are widely adopted compared to other Web 2.0 applications for educational purposes. Moreover, the use of wikis and blogs in education has been investigated in (Duffy and Bruns 2006). The Web 2.0 trends are applied to e-learning through courses that are application services, where an effective course should follow the principles of good Web publishing, the pages should be displayed easily without the need of special plug-ins and with appropriate navigation, and any integration to other services should be transparent (Rosen 2006).

Weller in (Weller 2006) predicts the future directions of learning environments and suggests that they will be based on Downes' Web 2.0 which has been described earlier (Downes 2005). The portal concept, which allows the user to tailor the content to their needs or preferences, is taken to the extreme in the e-learning 2.0 environment, where users can easily aggregate learning objects and all kinds of learning content from various sources. This leads to the environment presented as an *e-portfolio* tool (Walczowski and Ellis. 2007). One example of those new emerging e-learning Web-based applications can be seen in an application known as is *My Happy Planet*[30], which teaches languages through a partner who speaks the desired language to be learned. This partner is also interested in one of the languages the first learner speaks. Therefore, students learn languages by sharing their own knowledge of another language in return.

However, despite all the benefits of the new user-centric nature of Web 2.0 and e-learning 2.0, educators, teachers and learning institutes which choose to apply those technologies are faced with some new challenges (Millard and Essex 2007). For example, in a wiki, there is room for vandalism or false entries that could be presented without notice, especially as the idea behind the wiki is that what the majority thinks is right is so, which is not necessarily approved by all specialists or educators. One way to tackle this problem is by teaching the students themselves to use critical thinking and to question whatever they read or see, a skill that is much needed in life outside the classroom. Another issue is that as users are becoming increasingly diverse, especially in an online distance learning course or environment, they might have very different views that they

[30] http://myhappyplanet.com/

want to express; some would contradict others, and therefore learning how to deal with those situations is also important. An additional challenge is to know how to balance innovation and consistency when it comes to students' expectations and assessment. New methods and ideas could indeed enhance the learning experiences provided that they neither confuse the learner, who is overwhelmed by information. nor overshadow the original learning objectives (Millard and Essex 2007).

Finally, Web 2.0 tools should be used when they are needed and when there is an educational purpose behind that and not simply because they exist. People's needs should guide the technology and not *vice versa* as often happens in the case of e-learning.

3.3 Learning Management Systems (LMS)

Learning Management Systems (LMS), which are used as the content-delivery service in the research presented in this thesis, are integrated systems that support the needs and requirements of a number of teachers, administrators and students. A LMS describes software tools designed to manage users' learning interventions. It goes beyond conventional online training because of the extensive range of complementary functionalities, activities and tools that it provides to support the learning process. It aims towards a Web-based learning experience, which promotes the *anytime, any place* and *any pace* access to learning content and administration.

Teachers use LMS to provide courses, quizzes or assignments, communicate with their students and monitor the progress of an individual or the class as whole. Administrators use them for management and monitoring tasks, while students use LMS for the purpose of learning, communicating, sharing and collaborating with their peers or with the teaching staff.

LMS are based on different development platforms and languages; some are built in Java, C++ or PHP, using Java EE, Microsoft's .NET and other platforms. They have Graphical User Interfaces (GUI) and a robust database back-end to store learning content, learning data, users' information or users' grades and so forth. In addition to content creation, linking or storing, they provide tools to manage users, roles, courses and instructors and to facilitate and generate reports. Moreover, a modern LMS supports online assessment and can display students' scores and transcripts as well as track student activities.

Another term that comes across with LMS is LCMS, which stands for Learning Content Management System. The difference between a Learning Management System and a Learning Content Management System (although the term LMS is currently used to refer to both) is that LCMS mainly focuses on providing tools for authoring, re-using or re-purposing learning content. The primary user for a LCMS is a course developer or instructor, while the primary user of an LMS is the learner.

There has been competition between both the commercial and open source LMS. Some of the most popular LMS are Blackboard[31] (the former WebCT[32] LMS has become part of Blackboard) and ANGEL[33] which are all commercial products, while Moodle[34],

---

[31] www.blackboard.com

[32] http://www.Webct.com/

[33] http://angellearning.com/

[34] http://www.moodle.org

Sakai[35] and ATutor[36] are examples of the free and open source LMS. The research presented in this thesis used two open source products and therefore more attention is paid to open source LMS in the following sections.

Some independent third party surveys and studies (Adkins 2005; Aberdour 2007) have suggested that open source LMS will eventually dominate the market and might even replace the commercial products. This is due to a number of reasons such as the open source LMS' high scores in terms of innovation and advancement driven by their larger community of developers and users. The benefits of those LMS is clearly their lower initial costs, vendor lock-in avoidance, enhanced reliability, ability to tailor solutions to the customer's needs, localisation for a large number of languages, rapid fixing of bugs, continuous prompts for security patches releasing, and lower risks than the commercial LMS in terms of vendor collapse or product discontinuation. However, the surveys also considers the opinions of others who acknowledge the great potential of open source LMS, but believe that they are not ready to replace the commercial products, suggesting that those two types are more likely to co-exist. Reasons that are considered as limitations of the open source LMS are given by such views, such as extensive use of third party extensions that would raise the maintenance cost. In additional, the need for a local trained group to deploy and maintain the system would result in additional costs while in a commercial LMS this might be included in the license. Other issues raised by David Grebow[37] in (Adkins 2005) are what he has considered as bad graphical user interface GUI, lack of decent documentation and endless lists of features creeping in. The

---

[35] http://sakaiproject.org

[36] http://www.atutor.ca/

[37] Chief Learning Officer of Comcource: http://www.comcourse.com/index.html

customers do expect high quality, but in addition they are interested in lowering the costs, and existing customers will only switch from a commercial product to an open source one if they perceive a higher degree value, which in a demanding environment is a matter of innovation.

LMS in general (commercial and open source) have developed rapidly during the last few years, becoming more learner-centred and promoting collaboration, communication, knowledge sharing and support between both the learners themselves and the learners and their teachers in a democratic and non-hierarchical manner. Many tools have been developed and supported to achieve this goal. However, one of their limitations is that a typical LMS does not provide adaptation and personalised learning. It provides the same learning materials, which are usually stored in a local database, to all students regardless of their knowledge, goals or interests, without any buffered personal support (Brusilovsky 2004b). Lacking this important feature could limit the usefulness of this strong and powerful learning tool (Brusilovsky 2001a).

When deciding to use an LMS as a delivery service for adaptive learning, or in other words equip an LMS with adaptation features, it has been chosen to use open source LMS. Reasons included the low costs, community support and modular environments with open APIs which allow for easy integration. A number of features were suggested in (Avgeriou *et al.* 2003) when selecting an open source LMS in general, such as having an open source approved license, an active development community, stable versions and releases and be SCORM-compliant (chapter 2, section 2.7). In addition, it is important to have third party reviews of a particular LMS published. Moodle, which has been suggested for adaptation delivery by other studies as in (Graf and List 2005), as well as

ATutor, have satisfied those requirements to a large extent. Both systems are developed using PHP, MySQL and Apache software, which are widely supported. Moodle is introduced in section 3.3.1 while ATutor is presented in section 3.3.2.

3.3.1 MOODLE

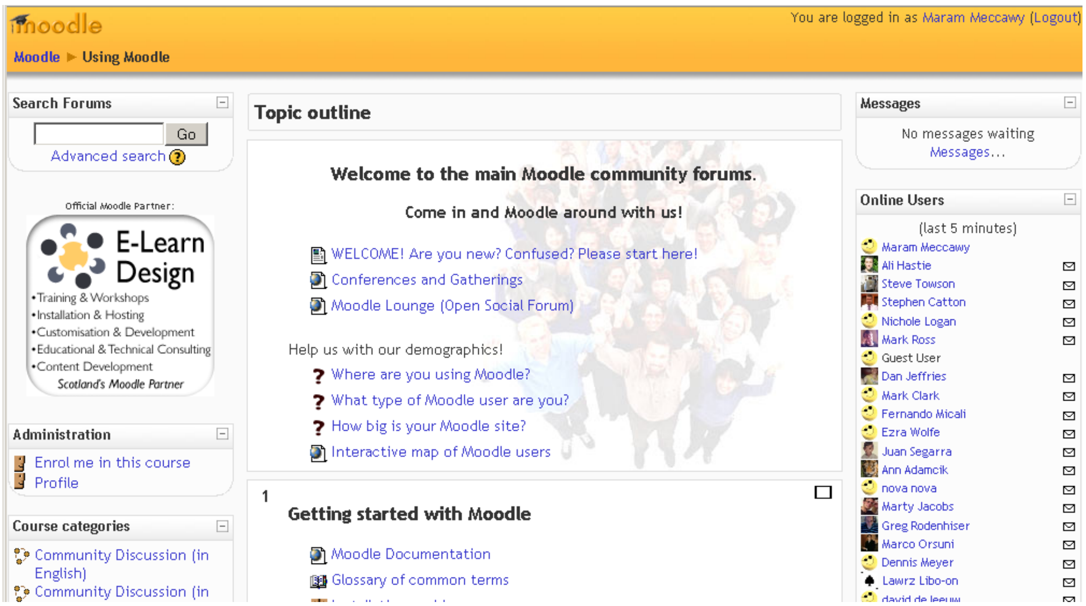

Figure 3.1: Moodle Learning space as it appears in the course *Using Moodle*

Moodle (Moodle 2008) is a Course Management System (CMS), Learning Management System (LMS) or Virtual Learning Environment (VLE). Many reasons lead to the choice of Moodle; it is a free and open source system which has a well-written and organised documentation, a large community of users all over the world and it supports 78 different languages. It emerged in 1999 and gained a wide adoption in the last few years, which could have been influenced by the UK's government-funded *OSS watch*[38] announcing that 56% of UK further education uses Moodle. In addition, the *Open*

[38] Open Source Software Advisory Service, http://www.oss-watch.ac.uk/

University[39] publicised its 5.6m *OpenLearn* initiative making 900 hours of learning content available to its students through Moodle's platform (See in (Aberdour 2007)).

Four main educational and philosophical concepts (Moodle 2008) were taken into account in Moodle's design, which have given Moodle its credibility and value; those are:

1. *Constructivism*: this point of view maintains that people actively construct new knowledge as they interact with their environment.
2. *Constructionism*: asserts that learning is particularly effective when constructing something for others to experience.
3. *Social Constructivism*: this extends the above ideas into a social group constructing things for one another, collaboratively creating a small culture of shared artefacts with shared meanings; hence it supports e-learning 2.0's concepts.
4. *Connected and separate:* a separate behaviour is when someone tries to remain objective and factual, and tends to defend his/her ideas using logic to find holes in their opponent's ideas. A connected behaviour is a more empathic approach that accepts subjectivity, trying to listen and asking questions in an effort to understand the other point of view. In addition, a constructed behaviour is when a person is sensitive to both of these approaches and is able to choose either of them as appropriate to the current situation.

In Moodle, there are many levels of management, which are handled by different types of users who have different roles defined by the system's administrator: administrators (site, course and user management), teachers (user and course management) and students

[39] http://www.open.ac.uk/

(user management). One user can be assigned more than one role that would allow him/her more capabilities and rights.

A typical Course in a standard version includes the following activities in addition to the course's text sections, those are: Lesson, Forum, Journal, Quiz, Resource, Choices, Survey, Assignment, Chats, Workshop, SCORM and Wiki. Moreover, other Moodle developers have added new activities, such as the Book activity. Those activities are organised around topics and courses. In addition, Moodle allows the addition of resources, which include composing websites or text pages, linking to external sites, adding IMS content package, inserting a label or listing a directory.

In addition, Moodle provides a teacher or administrator with a *user tracking* feature to track users by producing activity reports for each user. Nevertheless, it allows for some simple personalisation on different levels. One example is personalising calendar events. Also, the system provides some very basic semi-adaptation where no user model exists. Each lesson could end with a question and depending on the chosen answer the learner is led to one of three possible paths: either back, next or do not move. Alternatively, the teacher can remove the question and allow only one possible path, which is next (*i.e.*, provide the same lesson's content for every one).

Moreover, some of Moodle's features support Interoperability. For example, in the Quiz activity, interoperability is achieved by allowing the import of questions, problems or exercises from other systems. Moodle recognises: (AON, Blackboard, Course Test Management, GIFT, IMS, QT1, Missing Word Format and WebCT). In addition, Moodle allows for a SCORM–compliant course to be uploaded to the system as one of its activities, as mentioned earlier.

However, in (Adkins 2005) two problems, which will hinder the rapid uptake of Moodle, are described: lack of competency development features and poor user interface design compared with some other visually rich LMS rivals.

Two of the main Moodle components that were used heavily in the integration process (chapters 7-8) are: Quiz and Lesson (Cole and Foster 2005) activities, which are explained briefly below. In addition, the Book activity is also described because it has been used as part of the case study tools, although not in the adaptation context.

- **Quiz**

Feedback on students' performance is a critical part of modern education and seen as an essential feature that should be provided by an online learning environment. Moodle's Quiz is one of the most complex tools of the system. Moreover, a large number of options have been added by the community to the Quiz's engine to allow for flexibility and varieties. Quizzes can be created from different question types and they could also be generated randomly from a pool of questions. Moreover, students can be given the choice to re-take a quiz and have all those attempts scored by the system. Also, teachers can make the systems present immediate feedback associated with each right or wrong answer, which explains the obtained mark. Here, the added value of computer-based learning is demonstrated, since in a traditional classroom allowing a student to retake a quiz or test and then re-score it or re-mark it is a very tedious, time consuming and expensive process that learning institutions and teachers would normally avoid. On the contrary, this is a very simple process when using Moodle.

- **Lesson**

Like the Quiz, a Lesson activity is also considered one of the interactive, complex and very useful tools of Moodle. A Lesson can have a page or a number of pages, which can end up with a question and the navigation through those pages depends on the student's answer. This allows the teacher to create branching paths through the material based on the selections students make at each page. It could be seen as a simple form of adaptation or personalisation since the path of a right answer may vary from that of a wrong one, providing the learner with more support or challenges, but there is no user model. Alternatively, a flash-card lesson and branching tables could also be created. Hence there are two types of pages: 1) the *Question page* which ends up with a question and has the ability to be scored and this score is then added to the student's cumulative grade. 2) The *Branch table page*, which presents the student with the option to choose a branch. There are no correct or incorrect answers here for each response, and there is no impact on the students' grade depending on the choice he/she has made.

- **Book**

The Book activity or module makes it easy to create multi-page resources with a book-like format. It can be used to build complete book-like websites within a course. Moreover, previously-created websites can be imported directly into the Book. The book module allows the user to have main chapters and sub chapters only; the sub chapters them selves cannot have sub chapters. Since the book is intended to be a simple resource for teachers and students. Although the Book activity is not interactive, other resources and activities can be linked from within the book to add this interactivity when it is required.

3.3.2 ATUTOR

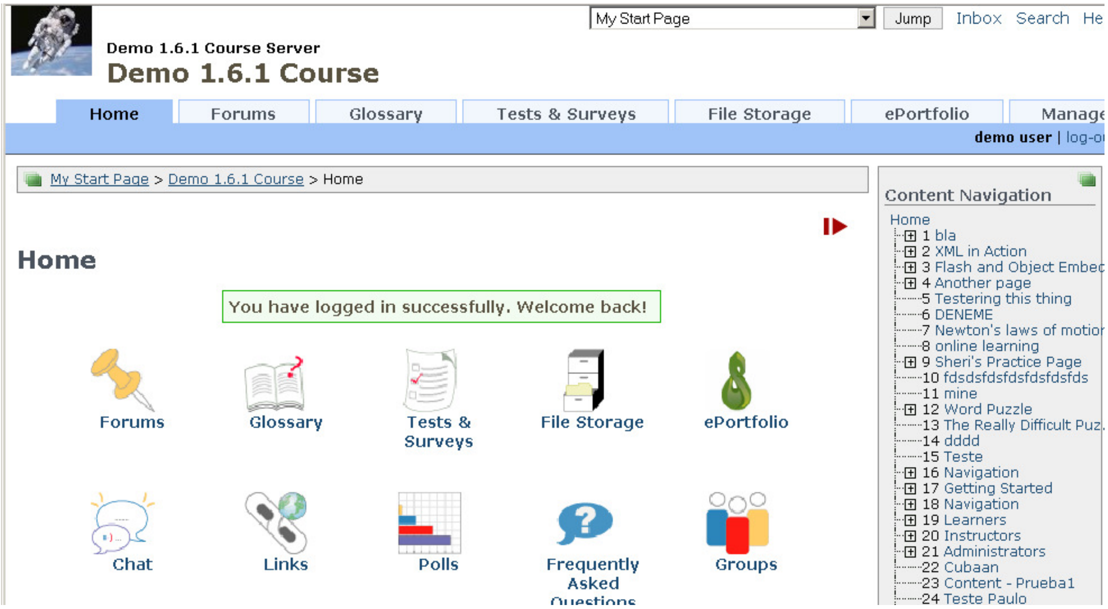

Figure 3.2: ATutor’s Course Demo

ATutor (ATutor 2008) is a free and open source Web-based LCMS designed to be used as a learning and teaching tool, which places great emphasis on standards and W3C compliances. Communities of developers across the globe have helped in developing it, including people from Canada, the USA, Italy, Norway and many others. A core team of 12 staff handles its development, which is supported by 16 separate organisations (Aberdour 2007).

The main approaches when developing the system were based on free software values, in addition to having ease of use, accessibility and authoring features combined in a modular architecture. Moreover, it has the possibility of extending its features by allowing the installation of additional (independently-developed) modules, hence promoting the creation of new modules and learning content as well as catering for reusability. In addition, it provides supportive student tools such as Chat, Forum, Survey and others. Those are combined together around the technology in tabs.

ATutor has a simple, well-organised and easy to use interface, which is built while focusing on accessibility, and which conforms to W3C accessibility guidelines as mentioned earlier. The interface is composed of a number of tabs and arranged in a group of features that are logically related. In addition, the navigation system throughout the system and within the content itself is simple and straightforward. Therefore, it requires no training and is accessible to people with disabilities. Moreover, this tabbed and hyperlinked structure of ATutor allows students to have an immediate broad overview of the site's navigation and arrangement. More description about some of ATutor's special tabs and hyperlinks (figure 3.3) are explained.

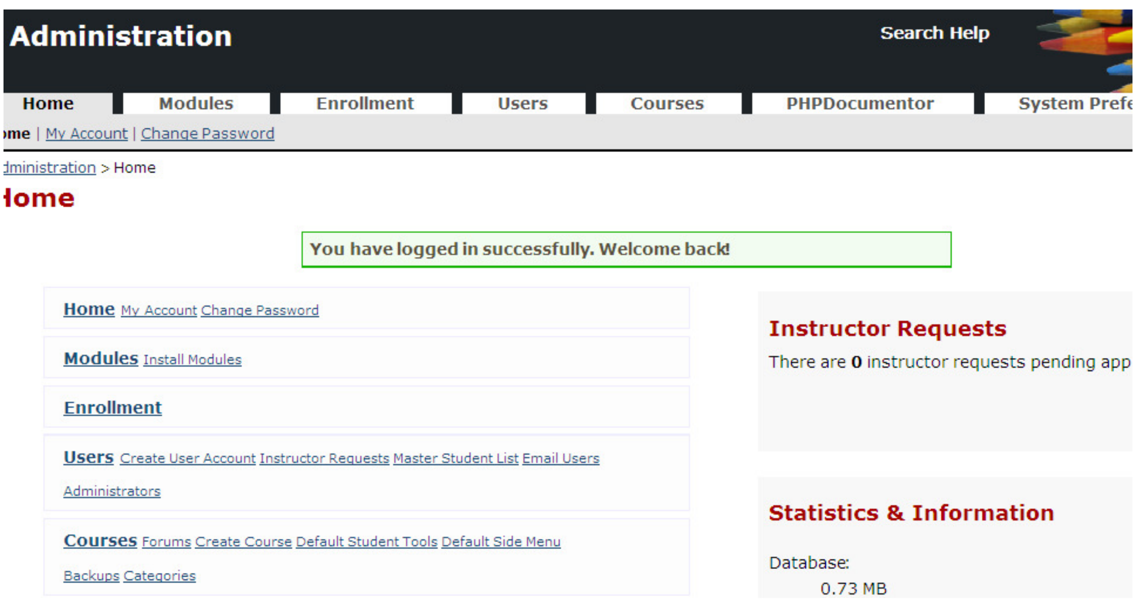

Figure 3.3: ATutor: Tabs and Hyperlinks.

- **HOME Tab**

The first tab displays a list of other menus accessible with tabs and the content of its options. This includes: My Account and Changes password options that are divided into two pages despite their simplicity. This is done from an accessibility point of view since tasks divided into shorter sequences could make it easier to use. As has been mentioned earlier in this section, simplicity and accessibility are in the main guidelines of ATutor's development.

- **MODULES Tab**

ATutor was designed to be modular: a variety of independently-developed modules could be located on the system's Website[40] or could be created by developers to suit their own needs. This tab provides a menu that presents users with options to install modules which in turn have their own options. One example of a module that a course provider might want to add is a payment module for purchasing this online course.

- **COURSES Tabs**

This menu is divided into a number of sections: Forums, Create Course, Default Students Tools, Default Side Menu, Backup and Categories. The *Create Course* sets up the course where the individual lessons can be added and later accessed using the side-bar navigation links.

- **DEFAULT STUDENT TOOLS**

Tools could be added to support the course provided that the teacher has a clear objective about what is to be achieved by using them. Those tools include: Forums, File

---

[40] http://www.atutor.cs

Storage, Chat, TILE Repository Search, FAQ, Links, Tests & Surveys, Sitemap, Export Content, My Tracker, Polls, Directory, Groups, Reading List and Blogs.

3.3.3 Moodle and ATutor

Moodle and ATutor have been selected to use for adaptive content delivery due to their advantages and capabilities. These two LMS have their similarities as well as differences and each one scores better in one or more feature when compared to the other. Table (3.1) provides a summarised comparison between. Moodle and ATutor. The focus here is on three aspects of the systems: technical flexibility, usability and general features as will be seen in the following table:

<table>
<tr><th>Categories</th><th>Subcategories</th><th>Moodle 1.8</th><th>ATutor 1.6</th></tr>
<tr><td rowspan="2">Technical Flexibility and Financial Requirements</td><td>Hardware and Software Requirements</td><td colspan="2">Apache/Microsoft IIS, PHP, MySQL/PostgreSQL, Requires only One database, Available for Linux, Unix and Windows</td></tr>
<tr><td>Cost</td><td colspan="2">Free & Open Source GPL</td></tr>
<tr><td colspan="2">Default Learning Tools
(Individual & Collaborative)</td><td>Lesson, Forum, Journal, Quiz, Resource, Choices, Survey, Assignment, Chats, Workshop, SCORM and Wiki.

Organised or grouped around topics/ courses</td><td>Forums, File Storage, Chat, TILE Repository Search, FAQ, Links, Tests & Surveys, Sitemap, Export Content, My Tracker, Polls, Directory, Groups, Reading List and Blogs.

Organised or grouped</td></tr>
</table>

			around the technology
Usability	**Installation**	Fairly easy	Very easy
	Materials	Authored or uploaded	
	Authoring Templates	Easy interface for creating content, very simple WYIWYG HTML editor	Allows both using HTML editor or manually code the HTML
	Standards	Could import SCORM 1.2 or AICC compliant learning content package	W3C WCAG 1.0 & WCAG 2.0 accessibility specifications, W3C XHTML 1, IMS/SCORM Content Packaging & SCORM 1.2 Runtime Environment (LMS RTE3)
General	**XML content metadata**	No	
	Languages	78	26
	Educational background	Based on four social & educational theories (see 3.4)	-
	Student Tracking	Yes	

Table 3.1: Moodle & ATutor Comparison Tables: Similarities and Differences

3.4 Fostering Adaptation within Learning Management Systems

Adaptive Educational Hypermedia Systems (AEHS) (chapter 2, section 2.3) suffer from a number of limitations as mentioned in the previous chapter (section 2.4). Some of these limitations include: lack of integration, limited support for functions and tools available in Web enhanced learning and limited reusability (Graf 2005). The opposite is true for the majority of LMS, which on the other hand suffer from lack of adaptivity or rich personalisation.

To get the best of both worlds, some AEHS researchers are proposing integrating adaptation with current popular and widely used LMS. Moreover, research studies which integrate other (non-adaptive) e-learning services, functionalities or systems with LMS have also been introduced. Some of those have focused on using e-learning standards such as SCORM, while others have provided various research solutions using technologies such as Web services, as will be seen in the following examples. Research presented in (Abu-Shawar 2006) describes the integration of the Student Information System SIS with Moodle at the Arab Open University (AOU)[41]. While having two separate copies of the database for LMS and SIS, yet maintain the synchronisation between those two. Another study has suggested the use of Web services to allow loose coupling and transparent integration of an LMS with third party, often legacy, application (Friesen and Mazloumi 2003). Here, the external resource is not controlled by a LMS; the LMS accesses the resource at a service level. These external resources can be any additional application that is made accessible and which has its own internal flow of control.

As for fostering adaptation features in LMS, research that integrates adaptation according to learning styles with LMS Moodle is described in (Graf 2007) and therefore will be elaborated upon here. It presents a meta-model for adaptive courses, where the content objects (text, pictures, links, multimedia files), have been tagged with meta-data that is used later for personalised selections. Here, Moodle have been chosen for adaptation delivery after studying a number of open source LMS to find which is more

[41] http://www.aou.edu.jo

capable of hosting the adaptation. In four defined categories: adaptability, personalisation, extensibility and adaptivity, Moodle's performance took over its rivals (Graf and List 2005). To enable this integration with Moodle, three main extensions have been made to its code. Extension one, extended the registration-form's code by providing and additional questionnaire to detect the students learning styles.. The second extension (expert model), extended the authoring tools to enable them to annotate the learning objects. Finally, the third extension (adaptation model), extended course- page's code to allow it to select automatically the right tailored version of the learning content as soon as the student logs in. However, this method which extends the code of a specific LMS is limited since the adaptation is specific for this particular LMS. What happens if the department or school decides to switch to another LMS? Therefore the solution could not be considered generic. More discussion about this issue in provided in chapter 9 (section 9.1).

Another work that intended to extend Moodle's functionalities to include adaptive testing framework is presented in (Sodoke *et al.* 2007). The aim was to improve e-testing by presenting students with questions that will take into account their different abilities and that might help in preventing cheating.

Moreover, a new European-funded project, launched in February 2008 and known as GRAPPLE[42] which stands for: Generic Responsive Adaptive Personalized Learning Environment, expressed a strong interest in using available LMS for providing adaptation. It states:

> *"To ensure the wide adoption of adaptation in TEL GRAPPLE will work with Open Source and commercial learning management system (LMS)*

[42] http://www.grapple-project.org/

developers to incorporate the generic GRAPPLE functionality in LMSs"(GRAPPLE 2008).

Other examples, which also aim to integrate adaptation with LMS, can be seen in the next section with the focus this time on group-level adaptation rather than on an individual-level.

3.5 Collaborative Learning and Group–Level Adaptation

There have been many definitions for Collaborative Learning, one broad definition presented in (Dillenbourgh 1999), which is relevant to the context of thesis states that:

> *"it is a situation in which two or more people learn or a attempt to learn something together"*.

It continues to explain that each of distinct elements mentioned in the above definition can be interpreted in different ways. *Two* can be interpreted as pairs, group, class, community or a society, *learning* may be interpreted as following a course, studying course materials or performing a problem solving activity, while *together* can be interpreted as face-to-face or computer mediated, synchronise or not, and there is no restriction for the frequency of the occurrence of this activity.

So far, adaptive learning was provided to students based mainly on their individual needs. However, learning is believed to be a collaborative process as much as an individual journey and the isolation of learners could affect the learning experience (Brailsford 2002b). On the other hand, as mentioned earlier, LMS have recognised such requirements for communication and collaboration in the online learning context and thus have provided the tools to allow for this collaboration to take place. Therefore, the research presented in this thesis integrates adaptation into an LMS to provide a combined environment that caters for both adaptive and collaborative learning. This has allowed the investigation of collaborative learning as part of the evaluation framework case study described in chapter 8 (section 8.5), and hence try to use the experiences gained from this study for the benefit of group-level adaptation (section 8.6).

The importance of group-based learning and its link to adaptive learning has recently started to concern the AEH community, resulting in a number of proposals to address this issue. However, it is worth mentioning that different groups view group adaptation in different ways. Some view it as adapting the adaptation methods and techniques (content, links, interfaces, or all) at the group's level using parameters that represent the group's characteristics rather than the characteristics of the individuals who form this group. This includes providing commonalities that lead to this grouping, for example stereotype groups. Others view group-adaptation as adaptation that occurs while students, who do not necessarily have any similarities, are working together to achieve a certain goal, or those who are actively participating in a given activity to complete a specified task, examples are given below.

In (Baajour 2007), the authors present an ontology-based approach for modelling the users and their interactions with the learning system using Web services to cater for interoperability. These lists of services with their application scenarios operate both at individual and group levels. Another study in (Papanikolaou *et al.* 2007), has investigated individual-to-group and group-to-individual influences and their possible applications in user modelling and hence adaptation. Moreover, Moodle is used to provide adaptation in the context of the feedback delivered to students to assess their progress in (Vasilyeva *et* *al.* 2007). It looked at feedback representation (what should be included), time (when to provide it) and the distraction to the learning process. The feedback can be tailored according to the characteristics of a specific learner leading to an individual-based adaptation. As for the group-adaptation part of the system, it is based on the characteristics of a group to which those individuals will be assigned according to the value of one or more parameters. Hence, the difference between the two types of feedback adaptation (individual-level and group-level) is in the way the user modelling and the user identification have been organised as well as how parameters were included in these models.

Research that has been explained in (Moreno *et al.* 2007) which also integrates adaptation into Moodle, has examined the individual level personalisation as the first step towards adapting the program to the group's level, suggesting many scenarios for adaptation when students are working together. Finally, research described in (Ounnas *et al.* 2007) suggests a semantic learner model based on the FOAF (friend of a friend) ontology (Brickley 2007), which is a vocabulary for mapping social networks. FOAF stresses that the automated process of grouping students while preserving the individual's personalisation, needs to be supported by an appropriate learning model.

Despite all the aforementioned potential benefits of group adaptation, it should not aim to replace individual adaptation since students remain unique and group adaptation could be looked at as another dimension for aiding students' needs. This has been also concluded by another study (Vasilyeva *et al.* 2007), which has suggested that a purely individual or a purely group-based feedback adaptation both have shortcomings, and therefore an approach that combines both types of adaptation is likely to be more adequate in providing useful feedback.

3.6 Summary

Moving from the advantages and limitations of AEHS in the previous chapters, this chapter provided an insight on the Web 2.0 concepts and technologies and their implications in the field of e-learning. This resulted in what is known as e-learning 2.0, where the user becomes the centre of the learning process. Moreover, the values of online socialisation, communication, contribution and sharing of knowledge are promoted in this new learning environment. However, there are some challenges will be presented to educators when using those modern technologies as have been mentioned earlier in this chapter.

A Learning Management System is the environment where modern e-learning takes place. Its great advantages and capabilities made it a justifiable choice for fostering personalised learning. LMS can overcome some of the AEHS limitations in terms of lack of integration, limited collaboration tools and prototypic interfaces. Moreover, they are widely accepted and used for delivering both online and blended learning courses. This chapter described two LMS that were heavily involved in the design, development or evaluation of the proposed architecture, namely Moodle and ATutor. Different research proposals for fostering adaptation and rich personalisation into LMS were analysed here. In addition, the chapter analysed different research experiments that integrated external systems/services with an LMS, which included adaptive and non-adaptive systems. Moodle was a very popular choice in different research projects, and Web services technologies have dominated such proposed solutions due to their capabilities which allow easy integration, loose coupling, interoperability and easy access. This is seen as a more flexible and efficient approach when compared to other approaches that for example extend the own code-base of the LMS, and hence require detailed knowledge of this code and limit the solution to only one specific LMS - the one which is currently in use.

Finally, a new level of adaptation, which is group-level adaptation, started to attract greater attention in the AEH community and some proposals have already emerged to address its issues. However, it should not be thought of as replacing individual-level adaptation; the two types should rather co-exist.

As mentioned earlier, Web services have many advantages especially when it comes to interoperability and the integration of different heterogenic systems as in case of LMS and AEHS. Therefore, they are used for developing the new architecture that will be presented in later chapters of this thesis. The next chapter provides a description of Web services and other software or technologies that are used in this development process.

CHAPTER 4

"Knowledge is of two kinds. We know a subject ourselves, or we know where we can find information on it"

Samuel Johnson (1709 – 1784)

SERVICE ORIENTED ARCHITECTURES AND WEB SERVICES

4.1 Overview

Interoperability, portability and flexibility are key aspects that have influenced the design of the new service-oriented architecture (presented in chapter 6, section 6.3). This architecture uses PHP, XML and XSLT to develop the components of the adaptive Web services which run over an *Apache* Web server[43]. The Web services are deployed and consumed following the SOAP approach using the PHP's *NuSOAP* toolkit. This chapter investigates the technical aspects of the architecture proposed by this thesis; it describes the platform upon which the architecture is built. In this way it lays the foundations before introducing the architecture in the coming chapters. Moreover, it justifies in detail the usage of those technologies as the project's implementation platform. Since Web services are the core building blocks for the new system, greater attention is given to this technology, as this chapter proceeds to demonstrate.

The Apache Software Foundation: http://www.apache.org/

4.2 The Platform

4.2.1 Why XML

eXtensible Markup Language or XML and its related languages such as XSLT[44], XPATH[45] and WSDL[46] are used to build the components of the architectural framework, as well as being the way of exchanging data between these different components and forming database repositories for the adaptive services. In addition, the educational resources were stored in XML files.

Many reasons influenced the choice of XML, such as its flexibility with data; any information can be placed into XML. It is an open standard which makes it even more flexible and generic since no one company or organisation has control over it. Furthermore, XML allows for Web page integration for data driven Websites (sites which customise their content according to user choices and selections) and it provides enhanced scalability and compression (POWELL 2007). All this serves the principles of the new architecture in being generic, interoperable, and portable. Also, in terms of the databases, XML provides not only data but data structure as well. Thus it contains metadata (data about data), which is much needed for the purpose of interoperability in e-learning systems.

Another key aspect of XML is that it is simple to use and understand and so passing XML documents between different databases, components, and applications or systems is relatively easy. However, pushing XML into another type of database such as MySQL, used by the majority of LMS including Moodle and ATutor (chapter 3, sections 3.3.1 and

---

[44] XSL stands for: eXtensible Stylesheet Language while XSLT stands for XSL Transformation

[45] XML Path Language

[46] Web Service Description language

3.3.2 respectively), is not a trivial task. Nevertheless, XML survives in heterogeneous environments but it might require the use of tools for specific databases or other standards for the translation or mapping. Hence, not only transferring XML itself but also sharing XML data and (ideally) automating the transfer of data between incompatible systems, would all eliminate the need of the custom coding (POWELL 2007). These capabilities resulted in XML being the language for Web Service technologies, which are services that provide some information over the Internet using a common, widely adopted transfer protocol such as HTTP.

An XML document is a database since it contains data and metadata which make it a repository of data, a very basic definition for a database. Using XML as a database has both its advantages and disadvantages. XML is object-structured and therefore capable of being used to build an object database. This object database has all the advantages of object-oriented techniques. These databases are much more capable of handling highly complex data structure when compared with relational databases, since an object breaks data down into smaller parts and the object design is therefore more flexible and efficient. On the other hand, XML databases become less efficient when dealing with a large amount of data. In other words, XML excels when dealing with complexity but not with scalability (POWELL 2007). Since the data size expected in the new architecture is relatively small (megabytes), the size problem was not an issue. Those XML-based databases were queried using XSLT, which is a stylesheet language that transforms XML documents.

4.2.2 PHP v4.3.1 and NuSOAP

The main reason behind choosing PHP is that the Learning Management System (LMS) in use - that is, Moodle - is developed and maintained in PHP. It would save time and effort to focus on one language to learn and use for the development process of the adaptation services, provided that this language works well in satisfying other requirements, which was the case with PHP. Nevertheless, given that the services are Web services, it does not matter which language was used for this development since everything is exchanged in XML.

PHP is a server-side scripting language that allows Web sites to be dynamic. Another feature that justified choosing PHP that it is a simple language with a short learning curve (NARAMORE 2005). Moreover, developing Web Services in PHP is made easy using specific classes and extensions freely available for users such as NuSOAP (Nichol 2004). NuSOAP is a group of PHP classes that allow developers to create and consume SOAP-based Web services. An additional good feature is its capability of creating WSDL description files (section 4.5) automatically. Moreover it creates an interface for that service that could be easily understood by ordinary users as in figure (4.1).

NuSOAP does not require any special PHP extensions, while such extensions are needed for using XSLT with PHP. In order to apply XSLT over XML using PHP, a special function called xslt_process (PHP 2008) is needed which in turn required installing additional PHP extensions. The XSLT module that comes with PHP must be enabled and for it to function properly, this module makes use of the open-source Sablotron XSLT processor. Sablotron, in turn, relies on the open-source XML parser, Expat (Yank 2001).

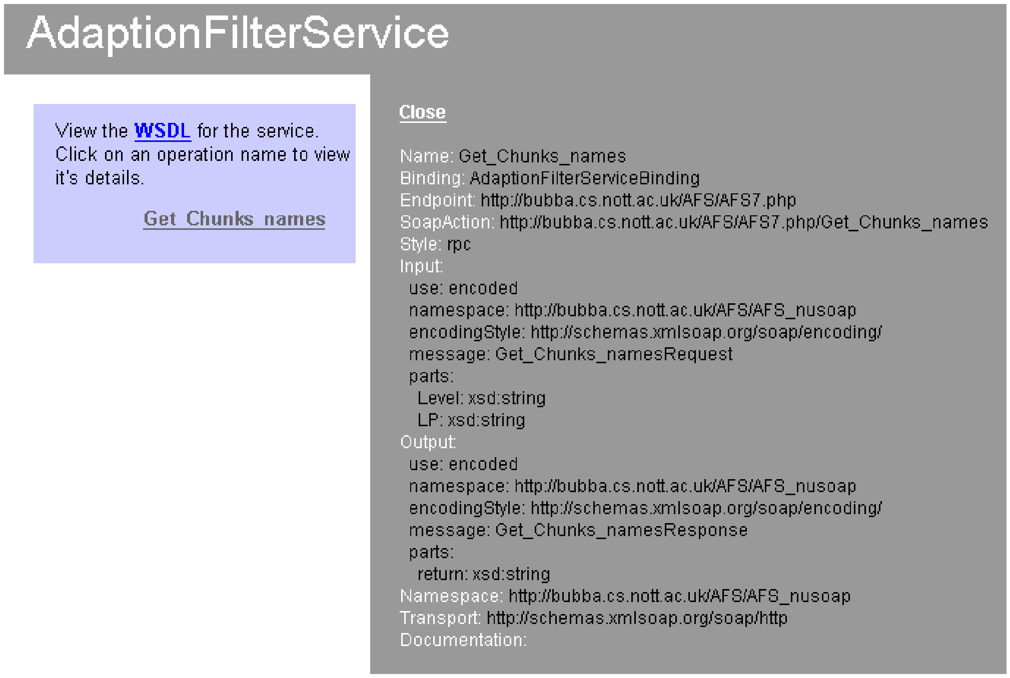

Figure 4.1: A Service Online Interface as produced by NuSOAP

The corresponding WSDL file is shown below in figure 4.2:

```
- <types>
  - <xsd:schema targetNamespace="http://bubba.cs.nott.ac.uk/AFS/AFS_nusoap">
      <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
      <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/"/>
    </xsd:schema>
  </types>
- <message name="Get_Chunks_namesRequest">
    <part name="Level" type="xsd:string"/>
    <part name="LP" type="xsd:string"/>
  </message>
- <message name="Get_Chunks_namesResponse">
    <part name="return" type="xsd:string"/>
  </message>
- <portType name="AdaptionFilterServicePortType">
  - <operation name="Get_Chunks_names">
      <input message="tns:Get_Chunks_namesRequest"/>
      <output message="tns:Get_Chunks_namesResponse"/>
    </operation>
```

Figure 4.2: A WSDL Interface as produced by NuSOAP

4.3 Distributed Computing and Middleware

In the early days of computers, the best solution for executing large-scale data was seen in using mainframe-based applications. Later, personal computers became available and applications running on standalone computers were very popular due to their low costs, ease of use and ownership privileges. However, as the number of computers and applications running on them grew, the communication between those applications became more complex, expensive and in some cases, impossible. Then networking computing gained wider importance enabling Remote Procedure Calls or RPCs over TCP/IP[47] protocols. Software running on a variety of hardware platforms, operating systems and networks were able to communicate with one another to share data. This high demand for communication and data transfer has led to the emergence of distributed computing applications (Kaye 2003). Distributed Computing is:

> "*a type of computing in which different components and objects comprising an application can be located on different computers connected to a network*."(Webopedia 2001).

In order for objects to communicate in a distributed environment, a set of guidelines or standards were needed. This resulted in a number of technologies such as CORBA[48] (1992), COM[49] (1993), DCOM[50] (1996) or RMI[51] (1996). Figure (4.3) gives a good example of distributed computing:

---

[47] Internet Protocol Suite: Transmission Control Protocol / Internet Protocol
[48] Common Object Request Broker Architecture (CORBA): http://www.omg.org/
[49] Component Object Model (COM): http://www.microsoft.com/com/default.mspx
[50] Distributed COM (DOCM): http://msdn.microsoft.com/en-us/library/ms809340.aspx
[51] Remote Method Invocation (RMI): http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp

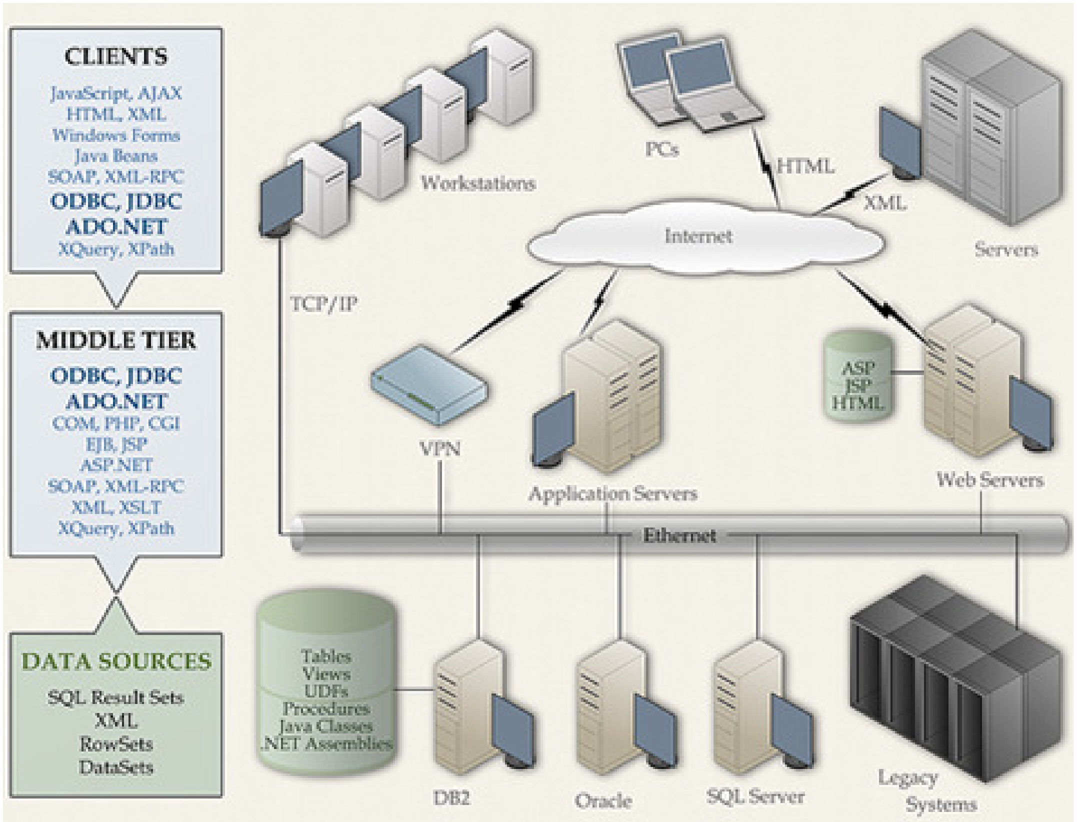

Figure 4.3: "A Multi-tier distributed application architectures provide performance and scalability. Data access middleware can operate from any tier of a distributed application". Taken from (Ken North Computing 2007).

Distributed computing has many advantages (Nagappan *et al.* 2003), which include its high performance since applications can execute in parallel and therefore distribute the load across multiple servers. Moreover, higher reliability and availability are provided by distributed computing since applications run on different machines. In addition, great extensibility and scalability in deploying reusable components on powerful servers are offered by this type of computing. Other advantages include high productivity, lower development cycle time (because problems are broken into smaller ones), reusability and reduced cost.

4.4 Service Oriented Architectures (SOA)

Services are the fundamental concept behind outsourcing: delegating jobs, tasks or responsibilities to other companies, partners and so forth, to do a job that formally one person, company or organisation would have done themselves. The reason for this delegation to other parties is either because it is more efficient to do so (resource-wise) or because they would deliver better outcomes. In the IT world, services are changing the fundamental model for software development and deployment; they continue to introduce new ways for building distributed applications where services are not to be connected with one another during the development process but also instantly during run-time. Service Oriented Architecture or SOA is an abstract description of how the different parts of the system interact and communicate to achieve a desired result (Kaye 2003). Webopedia's definition gives more attention to the interfaces stating that a SOA is:

> *"an application architecture in which all functions, or services, are defined using a description language and have invokable interfaces that are called to perform business processes."* (Webopedia 2006).

In its simplest form, a SOA could be described by the following diagram (figure 4.4):

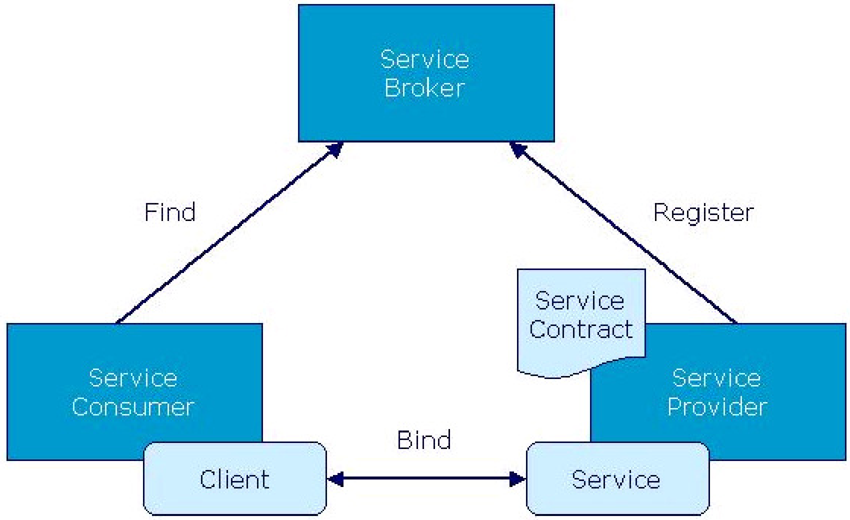

Figure 4.4: The Service Interaction Model, taken from (Strategic Planning Aligned Strategy 2006).

In SOA the architecture has three main players as can be seen in the previous figure (4.4): provider, discovery agency and requester which publish, find or interact with a service respectively. The communication between those parts could be done through interfaces or through messages.

The main differences between distributed computing and SOA is that SOA is a movement from the tightly-coupled, client-server model (2-tier) or even the client-middleware-server model (3-tier) into a peer-to-peer model where any node could be a client or a service. There is no special machine/mechanism to manage the network's resources; instead all responsibilities are informally divided among all machines (peers). The following figure (4.5) briefly states the difference between SOA and the 3-tier architecture:

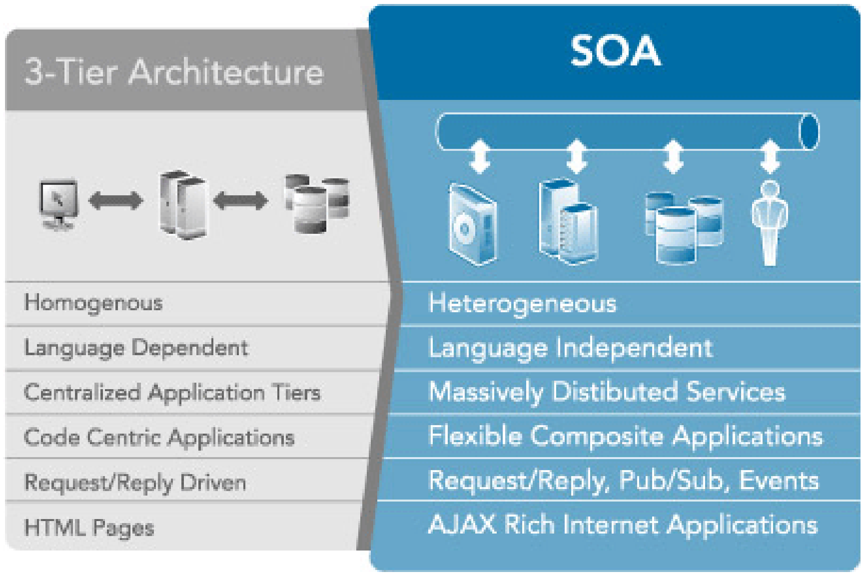

Figure 4.5: SOA versus 3-tier distributed architecture: *"Application Architectures are Evolving from 3-Tier to SOA"*. Taken from (TIBCO Developer Network 2008).

One of SOA's main concepts is *loose coupling*, which states that a communication could be established with a component, system or service without having to know its internal design or how it works. It is the opposite of *tight coupling*, where the programmers at each end must have substantial knowledge of components on both sides.

SOA is the foundation for Web services which are used as means of communication in the new architecture introduced by this thesis. Although SOA and Web services are used sometimes interchangeably, they are not the same. SOA is a design methodology or architecture, while Web services are the technology, an implementation of that architecture, which is not the only possible implementation for it. Web services are introduced in detail in the next section.

4.5 Web Services

4.5.1 Overview

A Web service is a software application which can be accessed remotely and that is identified by a URL like ordinary Websites. The difference between Web services and Websites is the type of interaction that they can provide (Potts and Kopack 2003).

Web services are a set of technologies and standards that allow a software developer to build distributed applications on top of the existing Web infrastructure and those applications use the Web as a transport layer (St. Laurent *et al.* 2001). Unlike client/server models, Web services do not provide users with a Graphical User Interface (GUI) since the applications communicate with one another and not with the users. It is for the developer to add a GUI if necessary. The level of application integration provided through Web services by allowing application-to-application integration has led to their rapid growth in popularity. Thus, some have argued that Web services are the next evolution of the Web (Beal 2005). Web services provide a solution to a problem that is faced in many areas when software systems seek to communicate with each other - that is, the problem of lack of interoperability. Interoperability is provided by allowing different applications from different sources to communicate with one another without time-consuming customised coding. Since all communications are in XML, the services are not tied to any specific operating system or programming language. Therefore, C++ communicates with Perl, Java with PHP and Mac with Unix or Windows.

There are many definitions for Web services; some are very narrow and others are broader. One definition in (Gustavo Alonso 2004) define a Web service as:

"an application accessible to another application over the Web".

Another definition which is more detailed and descriptive is provided in (Beal 2005) and states that a Web service:

> *"…describes a standardized way of integrating Web-based applications using the XML, SOAP, WSDL and UDDI open standards over the Internet protocol backbone".*

XML is used to tag the data; SOAP[52] is used to transfer the data; WSDL (Web Services Description Language) is used for describing the services available; and finally UDDI (Universal Description, Discovery and Integration) is used for listing what services are available. The use, therefore, of such open standards allows organisations to communicate with one another without having to know about the technology residing behind each other's firewall (Beal 2005).A similar definition for a Web service is announced by the W3C[53] (Gustavo Alonso 2004) :

> *"…a software application identified by a URI, whose interface and bindings are capable of being defined, described, and discovered as XML artefacts. A Web service supports direct interaction with other software agents using XML-based message exchanged via internet-based protocols".*

The following diagram presented in figure (4.6) provides a visual illustration:

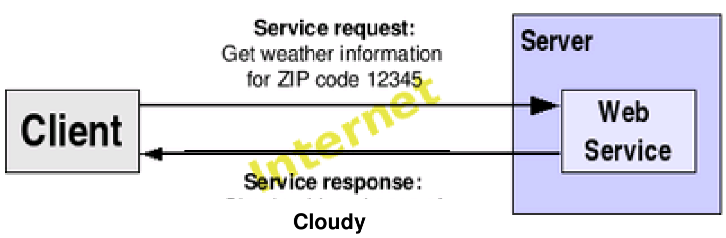

Figure 4.6 *" Web Services"*, adapted from (Sotomayor 2005).

---

[52] previously SOAP was an acronym for (Simple Object Access Protocol), this was ceased in SOAP 1.2
[53] World Wide Web Consortium : http://www.w3.org/

The power of Web services resides in the fact that each Web service implements a capability that is available to another or to other applications via standards, networks and protocols. They represent reusable software building blocks that are URL addressable (Chatterjee S and Webber J 2004). Such capabilities include functions, data, business process and many others.

A diagram provided in figure (4.7) puts the four Web service technologies in perspective as shown below:

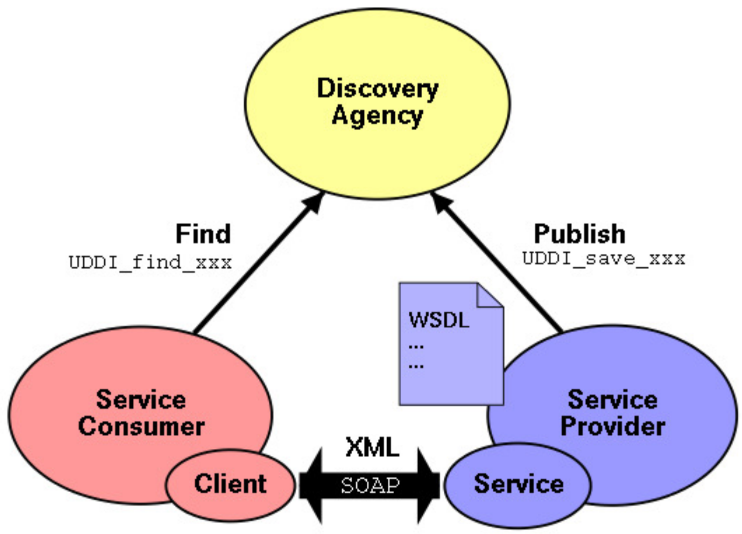

Figure 4.7 *"An Architecture View based on SOAP, WSDL and UDDI"*. Taken from (TIBCO Developer Network 2008).

The adaptive e-learning system that is proposed by the new service-oriented architecture is based on WSDL. WSDL is an XML document that conforms to a specification. All the services' metadata is contained at a certain location in this file and structured in such a way that will make it easy to understand. In addition, the WSDL file is human-readable, and therefore all that a programmer has to do to generate the code which is necessary to connect physically to services, is to use an XML parser to extract data into local variables.

This possibility of automatic code generation is one of the outstanding features provided by some Web services' toolkits such as in (MapForce 2006).

A WSDL document described in (W3C 2001):

> *"…defines **services** as collections of network endpoints, or **ports**. In WSDL, the abstract definition of endpoints and messages is separated from their concrete network deployment or data format bindings. This allows the reuse of abstract definitions".*

A WSDL document uses the following elements in the definition of network services: **Types**, a container for data type definitions using some type system (such as XSD[54]); **Message**, an abstract, typed definition of the data being communicated; **Operation**, an abstract description of an action supported by the service; **Port Type**, an abstract set of operations supported by one or more endpoints; **Binding**, a concrete protocol and data format specification for a particular port type; **Port**, a single endpoint defined as a combination of a binding and a network address; and finally a **Service, which** is a collection of related endpoints. This could be seen in the following diagram by W3C (figure 4.8):

[54] XML Definition schema

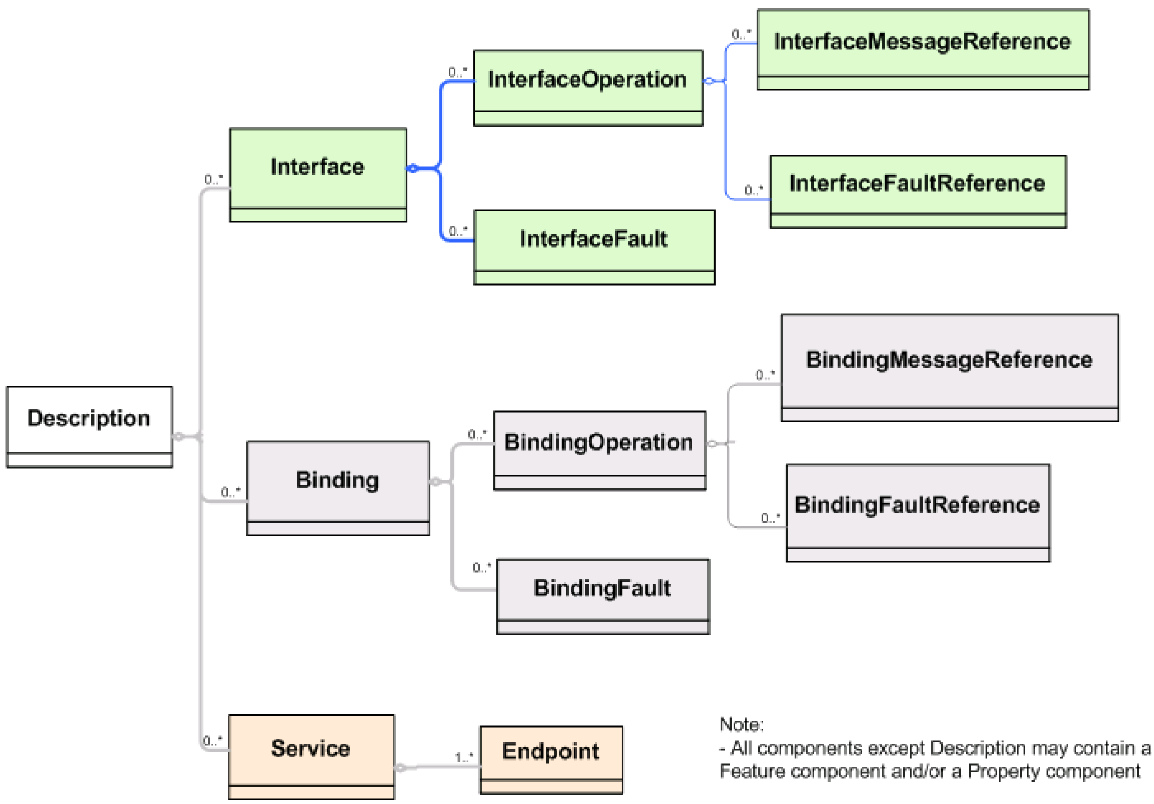

Figure 4.8 WSDL, taken from (W3C 2001).

WSDL is seen as a cornerstone in the Web services architecture because it not only provides a common language to describe the services but also provides a platform for integrating those services.

Web services use SOAP as a logical transport mechanism for moving messages between services described by WSDL interfaces (Chatterjee S and Webber J 2004). It is another important player in the Web service architecture. SOAP is an XML-based protocol for exchanging information between computers. Its purpose is to encode messages in a common XML format so that the message can be understood at each end (client and service). Figure (4.9) describes how a simple Weather Broadcast Web service could be invoked using those technologies which is more detailed compared with figure (4.6):

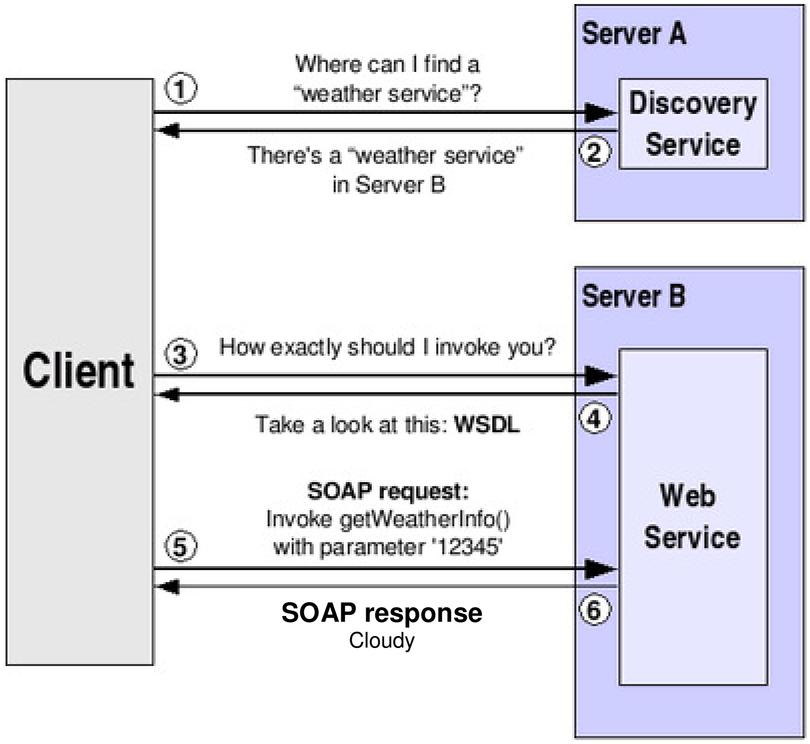

Figure 4.9 *"A typical Web Service invocation"*, adapted from (Sotomayor 2005).

However, SOAP is not the only approach to achieve this goal although it is the approach that has been followed in this thesis. Section (4.6) provides more detail about SOAP (6.6.3) and the two other approaches to deploy and consume Web services are: XML-RPC (4.6.1) and REST (4.6.2).

The Web service architecture could be therefore seen as composed of four main elements related to the service's processes, description, invocation and transport, which are best summarised in figure (4. 10):

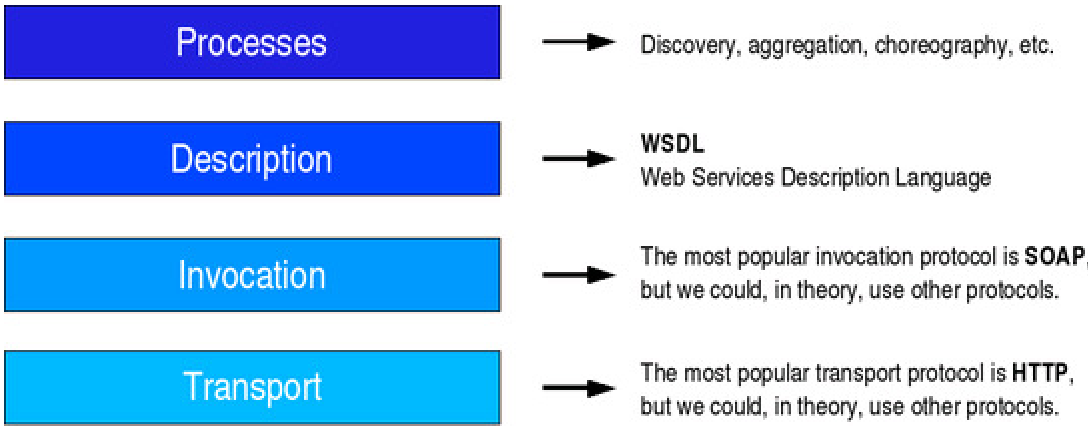

Figure 4.10 *"The Web Service architecture"*, taken from (Sotomayor 2005)

Amazon[55], e-bay[56] and Google[57] are examples of how Web services are used in the business industry.

4.5.2 Web Services Advantages Over Middleware Technologies

As mentioned earlier (section 4.3), a number of middleware technologies existed before Web services to allow distributed computing. CORBA, RPC, DCOM and RMI were designed to link the components of separate systems. However, they had their problems and limitations, such as being specific to the use of a certain development platform - for example, Java, in the case of RMI, or Microsoft in terms of DCOM. As a result, the community learned from those mistakes and later produced Web services. None of the former technologies were as generic as Web services and yet Web services owe a great deal to previous technologies (Kaye 2003). Another advantage of Web services compared to other technologies is loose coupling, which allows for flexibility and extensibility, while CORBA, DCOM and RMI are tightly coupled.

---

55 www.amazon.com
56 www.ebay.com
57 www.google.com

Web services resolved a number of problems that have not been solved by component-centric systems. Unique benefits (Kaye 2003) of Web services include: independence, standardisation, modularity and granularity, lower costs, reusability, loose coupling and scalability. Moreover, Web services do not require the same programs or systems to be written according to a certain programming architecture or style; object ordination, procedural or scripting languages could communicate with one another using Web services. In addition, Web services helped to leverage legacy systems since their great functionality could be exposed by wrapping their objects as Web services.

However, no technology is perfect; Web services in turn have their own limitations (de Jong 2002). Some believe that Web services have a high learning curve (Wright 2005) - this might be true if developers are building the services manually from scratch, but if they decide to use some of the available toolkits (such as PHP's NuSOAP section 6.2.2), the development, deployment and consumption of Web services becomes much easier. However, when comparing the overall performance of Web services with other proceeding technologies, it has been noticed that the Web services' performance does not reach their level. There is also a security issue, since Web services use the HTTP protocol which is privileged enough to pass firewalls, making it a potential threat. Nevertheless, the specifications of some new Web services have been introduced to deal with issues such as security, transactions and management (IBM developerWorks 2008). Table (4.1) briefly states the main features and limitations of each technology:

Technology	Features	Limitations
CORBA	- ORB (Object Request Broker) - objects from different systems cannot communicate - vendor-independent technology	- complex - not widespread - lack of extensibility - tight coupling - Requires the underlying application be Object Oriented because only OO is CORBA-compatible
COM	- objects within a single computer - Microsoft technology for Windows	- limited usage (local only) - vendor lock-in - costs - tight coupling
DCOM	- support communication between applications running on separate computers - Microsoft technology for Windows	- vendor lock-in - costs - tight coupling
RMI	- allows objects on one system to invoke methods on another possible remote system - Sun Microsystems for Java 1.1	- runs only on systems running Java written applications - tight coupling
Web Services	- vendor-independent technology - Independence - Self – describing and advertising - Standardisation - modularity and granularity - lower costs - reusability & Scalability - loose coupling - does not require programs to be written according to certain programming architectures	- high learning curve - poor performance when compared to other technologies - security issues (HTTP) - Some of the standards have not been yet approved

Table 4.1: Web Service compared with other proceeding middleware technologies

4.6 Approaches to Deploying and Consuming Web Services

4.6.1 XML-RPC

Remote Procedural Calls (RPCs) are much older technologies than the Web and in their basic form allow a developer to call functions across networks. RPC were developed by Sun Microsystems as a generic formal mechanism used for calling procedures and returning results over a network (St. Laurent *et al.* 2001). It was used extensively to facilitate and establish connections between two remote systems. Examples of popular RPC implementations include DCOM and CORBA (Jean-Luc 2004).

XML-RPC is a specification that comes from both RPC and the World Wide Web and it uses its own set of XML tags to mark up procedure calls. In addition, XML-RPC reuses another key component of the Web that is HTTP, which is the Web's transport protocol. HTTP itself is an RPC-based protocol with an open approach that sets standards for identifying and encoding different content, which has enough flexibility to carry the payloads demanded by XML-RPC (St. Laurent *et al.* 2001). The following figure (4.11) provides a conceptual description of XML-RPC:

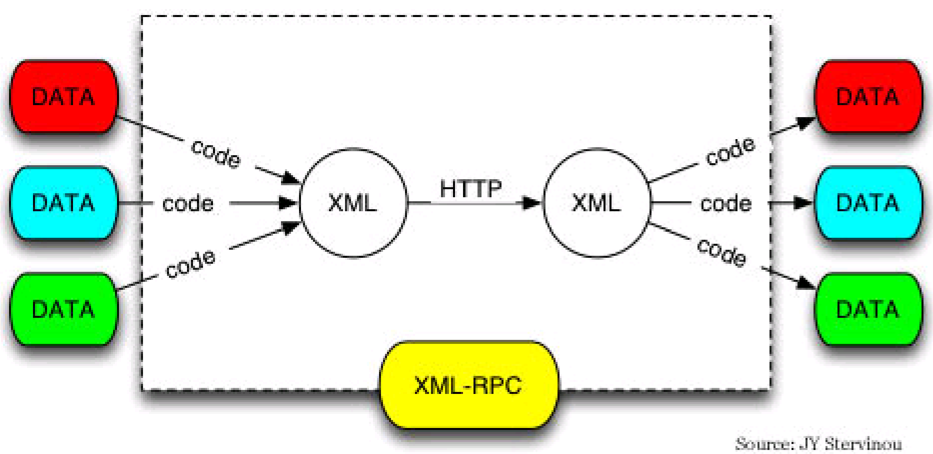

Figure 4.11: XML-RPC, taken from (XML-RPC 2003).

The main advantage of XML-RPC is that it is a simple and effective method of transmitting XML data (Jean-Luc 2004) ; one good example of its usage is Google APIs[58]. However, its relatively simple architecture limits its flexibility; it is too simple to perform some of the more complex functions. In addition, the relatively low level of abstraction can lead to potential complexity as the number of different requests increases. Moreover, some developers consider XML-RPC's reuse of the HTTP protocol conflicts with the infrastructure that supports this protocol. Security threats are a further issue, since XML-RPC over HTTP (which is effectively privileged) allows developers to bypass firewalls (St. Laurent *et al.* 2001). XML-RPC does not set out to solve every problem faced by Web services; it is concerned with simple, easy to understand requests and responses. It is therefore useful for standard data types and simple methods calls.

4.6.2 Representational State Transfer (REST)

Representational state transfer (REST), introduced by Roy Fielding in his PhD thesis (Fielding 2000), is another software architectural style for distributed computing such as Web services. It embraces a stateless client-server architecture in which the Web service is viewed as resources identified by their URLs (Tyagi 2006). Figure (4.12) shows the REST's structure:

[58] http://code.google.com/

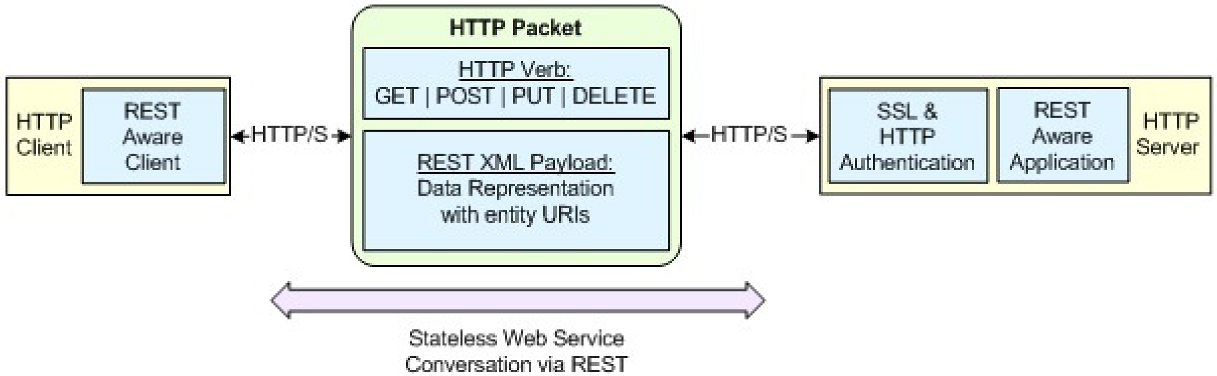

Figure 4.12: "*REST Web-Service Structure*", taken from (Hinchcliffe 2005).

REST's advantage is its simplicity since it uses HTTP methods such as GET, POST and PUT for sending and retrieving the data. The parsing must be done later using other tools. However, no tools, extensions or classes are needed to use it for data transfer. In addition, another advantage of REST is that it minimises latency and network communication while maximising the independence and scalability of components implementation, as described in (Fielding and Taylor 2000).

On the other hand, the disadvantages of this approach are that it needs to be done manually and hence requires thorough knowledge of XML and HTTP (Jean-Luc 2004). Nevertheless, when given the choice, developers seem to favour REST over the two other approaches because of its simplicity, as reported by both WS-Amazon and WS-flicker[59] (Dodds 2005), because its tight bound to HTTP and XML leaves them with only very little to learn.

4.6.3 SOAP

SOAP is a specification for XML-based distributed computing infrastructure (MapForce 2006). It is a stream of characters that are carefully created so that the

[59] http://www.flickr.com/services/api/

programs on both sides of the transmission can understand exactly what the other side is sending. Those characters are XML documents that are embedded in the transport's request and response messages. The W3C defines SOAP 1.2 as:

> *"…a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation specific semantics."* (W3C 2007).

A SOAP message is composed of three parts (figure 4.13), two of which are mandatory and a third which is optional. The mandatory parts are the SOAP envelope and SOAP body, while the optional part includes SOAP's header.

SOAP is designed as an XML wrapper for the requests and responses of Web services over the network. Its strengths lie in the use of namespace and XML data types in addition to its transport flexibility. It has a high level of abstraction, so that any operating system and programming language combination can be used to create a SOAP-compliant program. Further, although SOAP enables XML to be communicated over HTTP, it is not restricted to it; other options include using SMTP[60] or FTP[61].

[60] Simple Mail Transfer Protocol

[61] File Transfer Protocol

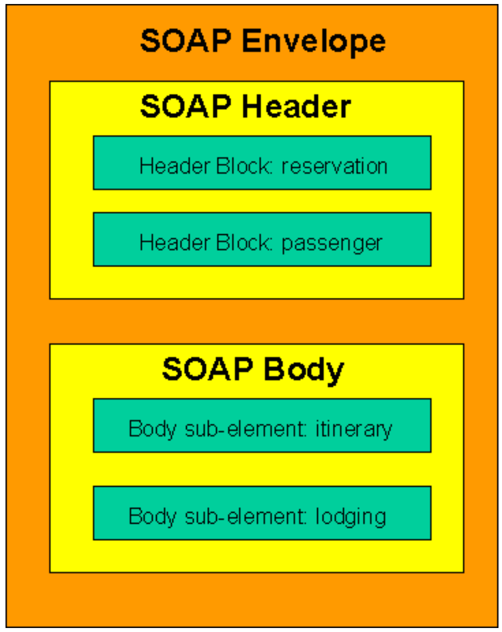

Figure 4.13: Structure of a SOAP Message, taken from (W3C 2006).

For this particular research which is presented in this thesis, a SOAP API for the PHP platform known as NuSOAP (section 4.2.2,) has been used to deploy and consume Web services. The availability of this tool for PHP is one of the reasons for choosing the SOAP approach. Moreover, the fact that by using SOAP the operations are defined as WSDL ports that are easily displayed and human-readable gave further encouragement to use this approach. In addition, other advantages include the possibility for debugging as well as hiding complex operations, and compared to the other two approaches it has increased privacy (Zur Muehlen *et al.* 2005). Google as well as Amazon[62] have both created SOAP-based Web services (Jean-Luc 2004). However, all these advantages come with a price: SOAP's implementation is more complex compared with the formal

[62] http://www.amazon.com/AWS-home-page-Money/b?ie=UTF8&node=3435361

approach. In addition, when using SOAP, a client needs to know operations and their semantics beforehand (Zur Muehlen *et al.* 2005).

4.6.4 Comparing XML-RPC, SOAP and REST

Each approach as seen has its own advantages and limitations, suggesting that depending on the system's requirements and/or the skills of the developer, a certain approach might be more appropriate than others. However, it is interesting to realise that although those technologies are set to deal with interoperability issues, they are not interoperable with one another as will be seen in the following table (4.2) - provided by IBM's developers - which compares the three approaches :

	SOAP	XML-RPC	REST
Definition	SOAP is a lightweight protocol for information exchange in a decentralized, distributed environment. It is an XML-based protocol that consists of three parts: an envelope, a set of encoding rules, and a convention for representing remote procedure calls and responses.	This is a remote procedure call that uses HTTP as the transport and XML as the encoding. XML-RPC is designed to be as simple as possible, while allowing complex data structures to be transmitted, processed and returned.	Representational State Transfer is intended to evoke an image of how a well-designed Web application behaves: a network of Web pages, where the user progresses through an application by selecting links, resulting in the next page being transferred to the user and rendered for their use.
Goals	SOAP extends XML-RPC by implementing user defined data types, the ability to specify the recipient, message specific processing control, and other features.	A clean, extensible format that is very simple. An HTML coder should be able to look at a file containing an XML-RPC procedure call, understand what it is doing, and be able to	REST was created to provide a design pattern for how the Web should work, and to serve as the guiding framework for the Web standards and

		modify it and have it work on the first or second try. An easy-to-implement protocol that could quickly be adapted to run in other environments or on other operating systems.	designing Web services.
Data types supported	Integer, Boolean, ASCII String, double precision signed floating point number, date-time, structs, arrays, array of bytes, enumeration, User-defined data types, polymorphic accessors	Integer, boolean, ASCII string, double-precision signed floating point number, date-time, structs, arrays.	Implementation specific. In general supported types are Integer, boolean, ASCII string, double-precision signed floating point number, date-time, set, list, properties.
Simplicity	Slightly complex compared to XML-RPC	Easy to understand and develop	Implementation-specific
Stability	Accepted standard under W3C	Not a standard	Architecture reference. Standard not needed
Interoperability	Cannot interoperate with REST/XML-RPC	Cannot interoperate with REST/SOAP	Cannot interoperate with SOAP/XML-RPC
Tooling	Many major companies including IBM and Microsoft started supporting SOAP in their tools.	Tooling is still evolving.	Not many tooling support is available.
Customizability	Highly customizable, data-types and protocol independent	Lightweight, can work only on HTTP, limited data type support	Works only on HTTP
Library	Many open source libraries are available.	Many open source libraries are available.	Not many implementation libraries available

Table 4.2: *Comparison of SOAP, XML-RPC and REST*, adapted from (Subbian and Kannan 2006)**.**

4.7 Multiple Web Service

The proposed architecture consists of a set of independent Web services which work together to achieve a common goal of providing adaptation of e-learning content. Therefore, there is a need for a defined approach for this collaboration to work effectively.

When discussing multiple Web services two terms are usually described: Orchestration and Choreography, which aim to provide a conversational model between different Web services.

According to Plez, while the two terms overlap, orchestration refers to:

> *"…an executable business process that can interact with both internal and external Web services. Orchestration always represents control from one party's perspective. This distinguishes it from choreography, which is more collaborative and allows each involved party to describe its part in the interaction".* (Plez 2003)

Collaborative interoperability relates to the meaningful interaction of services with other services, not just between a client and a service (Wright 2005). A number of approaches are available to deal with multiple Web services; these include the use of some existing or new initiatives, specifications or software products, such as: WSCI[63],

[63] Web Service Choreography Interface (WSCI) 1.0: http://www.w3.org/TR/wsci/

BPEL[64], ESB[65], WCF[66] and others. Those are different from traditional middleware solutions as explained by the W3C's WSCI specification:

> *"the loosely coupled, distributed nature of the Web prevents a central authority (or a centralized implementation of a middleware) from exhaustively and fully coordinating and monitoring the activities of the enterprise applications that expose the Web Services actually participating in the message exchange"* (W3C 2002b).

Another concept which is also related to multiple services is service *aggregation* where one service makes use of other services (Kaye 2003). An aggregator service acts both as a client and a service. A service, when responding to an initial request by a certain client and then - for example - performing a specific task (breaking a request into two smaller independent requests), becomes a client or a requester that invokes other services. The results of those invocations are then returned to the requester (the aggregator service) which in turn passes it back to the initial client. The aggregation approach has been used in the architecture proposed by this thesis to coordinate the communications between different Web services (see figure 6.4 in chapter 6).

The requests and responses between different services could be synchronous or asynchronous depending on the nature of the application. Although the aim of aggregating services is to simplify and coordinate the provided services for the requester (less request calls needed), in some cases this series of internal service calls could lead to some sort of complexity and produce new problems such as reliability, efficiency and

[64] Business Process Execution Language for Web Services: http://www.ibm.com/developerworks/library/specification/ws-bpel/
[65] Enterprise Service Bus: http://www.sonicsoftware.com/solutions/service_oriented_architecture/enterprise_service_bus/index.ssp
[66] Windows Communication Foundation: http://msdn.microsoft.com/en-us/netframework/aa663324.aspx

availability. What happens if one service fails? How would the aggregator service guarantee the quality of service (QoS)? Those are questions that need to be considered.

4.8 Summary

The chapter aimed to set the technical foundation for the e-learning architecture that is presented in the coming chapters. The proposed architecture is distributed in its nature and uses Web services as a means of communication. Therefore, this chapter provided an overview of some technologies and software architectures and styles such as distributed computing, SOA, Web services and its own technologies such as WSDL, SOAP and UDDI. In addition, other available technologies for transmitting data using Web services were briefly introduced: XML-RPC and REST.

Moreover, the implementation platform upon which the architecture is built was also introduced. XML's flexibility and openness as well as PHP's simplicity and short learning curve were among many desirable characteristics that resulted in their use for developing the new architecture.

Interoperability is one of the key advantages provided by Web services and is also one of the main requirements for the new architecture as explained in previous chapters. Therefore, the next chapter practically investigates the issue of interoperability regarding adaptive learning content in the context of Adaptive Educational Hypermedia Systems via the use of two case studies as will be demonstrated shortly.

CHAPTER 5

“A thinker sees his own actions as experiments and questions--as attempts to find out something. Success and failure are for him answers above all.”

Friedrich Nietzsche (1844 - 1900)

INVESTIGATING INTEROPERABILITY

5.1 Overview

Authoring content for adaptive hypermedia is a rather complex and time-consuming process. This remains the case until user-level authoring tools become widely available. However, although the cost for initial development remains high, reusing the existing learning content could help in lowering the cost by avoiding a situation where the same or similar content is created more than once. Research on interoperability and reusability of learning content in Adaptive Educational Hypermedia Systems (AEHS) has therefore attracted attention in the Adaptive Educational Hypermedia (AEH) community, as seen in chapter 2 (section 2.5). In order to propose a new adaptive e-learning architecture to address those issues, it is necessary to first investigate different approaches for achieving interoperability.

This chapter presents two case studies that were designed in the process of creating the new architecture that will be presented in the next chapter (6). They both investigated reusability of learning content by applying different approaches. The first case followed the conversion approach (chapter 2, section 2.5) between two AEHS but extended it by using the technologies of Web services (chapter 4, section 4.5). The second case study followed a non-standardised procedure that uses original communication protocols to achieve interoperability with the focus on rich and interactive content rather than simple content. This approach is classified under the Language-specifier approach that has been described in chapter 2 (section 2.5).

One interesting aspect of the research presented here is the involvement of different international research groups in those projects, which indicates the significant impact of research collaboration in achieving interoperability in this area.

Research described in this chapter was published in several papers in workshops, conferences and journals (see list of publications on page XII).

5.2 CASE STUDY 1: Interoperability Of Learning Content And The *MOT* To *WHURLE* Conversion, A Web Service Approach[67]

5.2.1 Introduction

---

[67] This study case was published in the following peer-reviewed papers (can be retrieved from http://www.cs.nott.ac.uk/~mzm):

Meccawy, M., Stewart, C., and Ashman, H., Adaptive Educational Hypermedia Interoperability and Content Creation with a Web Service-based Architecture. *International Journal of Learning Technology (IJLT).*2007. 7 (3): p. 269 - 285.

Meccawy, M., Celik, I., Cristea, A., Stewart, C., and Ashman, H. Interoperable Adaptive Educational Hypermedia: A Web Service Definition. In: Proc. 6th IEEE International Conference on Advanced Learning Technologies (ICALT 2006). Kerkrade, The Netherlands: IEEE Computer Society, July 5-7. 2006.

Meccawy, M., Stewart, C., and Ashman, H. Adaptive Educational Hypermedia Content Creation: A Web Service Based Architecture In: Proc. Adaptive Hypermedia (Ah2006) Workshop on Authoring for Adaptive Hypermedia. Dublin, Ireland, June 20. 2006.

There are many distinct AEHS, many of which have been developed and used in a research environment, such as AHA!(De Bra 2003), MOT (Cristea and De Mooij. 2003), and WHURLE (Brailsford 2002b). Each of these systems adapts to a slightly different *view* of the learner, for example, WHURLE stores each learner's knowledge level (a beginner, intermediate or advanced learner) at the granularity of each domain. Any teacher (assuming a non-technical expert who wishes to use an AEH system) who is considering authoring an adaptive lesson faces many decisions, such as:

- Which AEHS should he/she use?
- Does any previous material exist and is it in a usable condition?
- What are the objectives of the lesson and how are they to be achieved for a heterogeneous group of learners?
- How many versions of the same material need to be created?
- How are the various versions to be presented to the learner?

As a result, authoring for an AEHS is a complex and time-consuming process and so reusability of adaptive content is needed, which should be possible if AEHS are made interoperable with each other.

Research on generating a new authoring paradigm, one that moves away from the 'create once, use once' approach, already existed (Stewart *et al.* 2004; Stewart *et al.* 2005; Cristea 2005); it has suggested that only interoperability between AEHS will address authoring issues. When an author can develop learning materials in one system and then have them delivered in another, AEH will begin to enter the mainstream of education.

This case study recommended using a Web service-based approach to solve the interoperability issues further. An initial implementation between two AEHS - MOT and WHURLE - that builds upon previous conversion between these systems is presented here. The aim was to use this case study as a first step to learn more about the issues involved in creating a Web service that can convert from one system to another. It was the first step in creating a middleware system that can transfer adaptive learning materials transparently between systems.

5.2.2 The MOT to WHURLE Conversion

MOT is a generic Web-based AEHS based on the LAOS framework (Cristea and De Mooij 2003), for the authoring of adaptive hypermedia. It implements a simplified version of the LAOS layers (chapter 2, section 2.6). In this case study, only the transformations of the two models involved in this conversion has been referred to: the Domain Model, and the Goal and Constraints Model. GMs (MOT Lessons) are intended to be the basis of the AEH presentation in MOT, as they contain the pre-selected material appropriate for the goal and target audience of the presentation, hence the illustrations presented here will represent their conversion.

Using MOT, authors can create domain concept maps (DM) containing the actual learning content, and they can also create the lessons themselves (Goal & Constraint Maps, GM), based on these domain maps, that allow a restructuring and filtering of the learning materials. These contents are stored in a MySQL database.

As for the second element of this conversion equation - that is the AEHS known as WHURLE - its content model consists of two major layers. The *melange* consists of the content separated and distributed into *chunks.* A chunk is a conceptually self-contained unit of information. In WHURLE, chunks are either written by subject experts or imported from MOT, or converted from legacy data. The second layer of the content model is the *structural overlay* created by lesson authors as the default narrative of the student experience. Lessons are defined by *Lesson Plans* (LP) that provide a hierarchical framework for the appropriate chunks. A WHURLE LP can be created either from a) a MOT Domain Map, or b) a MOT GM. However, there is an important difference in that WHURLE has no conceptual equivalent to the separation of the Domain Model and the Goal and Constraints Model of MOT. WHURLE will be described in further detail in chapter 6 (section 6.2) because of its relevance to the new system described there, hence this would allow for a comparison between the two systems.

Enabling interoperability between two very different systems is not straightforward. The first step towards this has been taken in (Stewart *et al.* 2004) but this approach was limited. It enabled a direct conversion of learning materials using MOT as the authoring system and WHURLE as the delivery system. However the conversion program itself was an off-line, command-line based system. To implement transparently a 'create once, use often' paradigm for authoring, it would be more appropriate to create a Web-based system. Moreover, if the system is to be used by non-experts, then a GUI is needed for the conversion to become an easy task.

5.2.3 MOT To WHURLE Conversion: A Web Service Approach

In order to create a completely new Web service, the following stages are generally required in this process (from the perspective of a service developer): create core functionality of the service (system); create WSDL service description of that system; create a SOAP service wrapper (if using this approach for deploying and consuming Web services, see chapter 4); and deploy the service onto a server and register the new service via UDDI, where it could be found by other consumers. Since the conversion program was already available from previous research work, the first step in developing the service (creating the core functionality) was skipped, and this case study started by creating the WSDL file followed by the SOAP examples.

- **Defining the WSDL abstract layer**

In order to create the *WSDL semantic abstraction layer* that allows for AEH content conversion between the two separate systems (MOT and WHURLE), the commonalities between the two systems had to be identified and described. To describe these commonalities in WSDL, first the abstract section was created by defining the WSDL types <wsdl:types> for both MOT and WHURLE data types.

- **MOT Lesson conversion into a WHURLE Lesson Plan**

The adaptive MOT lesson conversion into an adaptive WHURLE lesson plan was done by extracting the basic elements from the two systems and re-composing them; figure (5.1) shows an extract of a MOT lesson.

```
<xsd: element name="MOTlesson">
    <xsd:complexType>
        <xsd:sequence>
            <xsd: element name="lessonID"  type="xsd:int" />
            <xsd: element name="ToplessonID"  type="xsd:int" />
            <xsd: element name="Sublesson">…
```

Figure 5.1: A snapshot MOT lesson in WSDL

The composition of WHURLE lesson plans, containing chunks, is in figure (5.2) below:

```
<xsd: element name="WHURLElessonplan">
<xsd:complexType>
    <xsd:sequence>
        <xsd: element minOccurs="1" name="level">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd: element minOccurs="1" name="page">
                        <xsd:complexType>
                            <xsd:sequence>
                                <xsd:element minOccurs="1" name="WHURLEchunk">
```

Figure 5.2: A snapshot WHURLE lesson plan in WSDL

- **Message, operations and PortTypes:**

Next, the main messages that are exchanged or transmitted during the conversion process are described. Figure (5.3) shows the messages that allow the user to select the lessonId for a MOT lesson to be converted.

```
<wsdl:message name="printLM">
    <part name="MOTlessonId"  type="xsd:int"/>
</wsdl:message>
<wsdl:message name="printLMRespons">
    <part name="returnIdWhichLM"  type="xsd:int"/>
```

Figure 5.3: WSDL Adaptive Lesson’s Conversion messages

Following the message description is the portType description which includes the operations (functions) that are supported within the conversion service. An example is given in figure (5.4), which describes the above messages as part of a single operation. This operation identifies the messages that need to be sent, and the message that can be expected in return.

```
<wsdl:portType name="MOT2WHURLE_Conversion_PortType">
    <operation name="Mlesson_Wlessonplan_conversion">
        <input message="tns:printLM"/>
        <output message="tns:printLMRespons"/>
    </operation>
```

Figure 5.4: A snapshot for the "Mlesson_Wlessonplan_conversion" operation

By describing the systems' data types, the messages, as well as the available operations within the portType, the abstract description of the WSDL file was completed. Next, the description of the concrete (physical) part, which includes the WSDL binding, port and service, is conducted. This part is responsible for connecting to the abstract part physical binding between a client and a service**.**

- **Binding, Port and Service:**

Binding, as mentioned earlier, is about how the message will be transmitted, and it has two main purposes: linking the abstract and concrete elements in WSDL, and as serving as a container for information such as protocol and address of the Web service. The WSDL file is completed by declaring the *port* and *service* elements, which identify the service's location.

- **Creating SOAP wrappers**

In order to test the service and exchange messages/information with its server, the SOAP messages and response files were created. These are XML files that, as described earlier (chapter 4, section 4.6.3), consist of three elements: Envelope, Body and Header (optional). Below is an example of a SOAP request in figure (5.5):

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:printLM
      xmlns:ns1="http://cs.nott.ac.uk/~mzm/MOT2WHURLE/"
      SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <MOTlessonId xsi:type="xsd:int">12</MOTlessonId>
    </ns1:printLM>
 </SOAP-ENV:Body> </SOAP-ENV:Envelope>
```

Figure 5.5: A snapshot of a soap request (printLM)

As can be seen from the previous examples, the user (Web service's consumer) entered an Id number for a lesson in MOT (here it is '12') to be converted into a WHURLE lesson plan (LP) (chapter 6, section 6.2) and the response message carried this number to the corresponding function in the conversion program (service), which resulted in displaying the equivalent lesson plan (LP) in WHURLE after the conversion process has occurred.

5.2.4 Discussion

This case study showed that Web Services hold promise for achieving interoperability, at least for its technical aspects. However most approaches for enabling interoperability focus on relatively simple forms of data. Yet, in adaptive hypermedia systems, not only are semantic differences an issue, but there is the problem of interdependency between parts of AEHS. It is not enough to enable interoperability at a technical or even semantic level, as the data and processes of the distinct parts of the adaptive system are by no means autonomous. In fact, interoperability is needed on three levels: technically, semantically and also on a structural level (Sun and Fu 2005).

Using the approach of the previous system described in (Stewart *et al.* 2004; Stewart *et al.* 2005), conversion software is required for each pair of systems requiring a conversion. In contrast, using an approach that creates a general framework for sharing materials removes much of the effort involved in implementing the conversion. Only a single "convert-to" and "convert-from" process must be defined for each participating system, rather than the same processes being defined for each pair of systems. This may not be a major issue, but becomes so when one wants to create materials for use in many other systems, or source materials from many other systems.

In applying a Web Service-based approach this addresses two problems with the initial conversion system: *centralisation* and *generalisation*. Centralisation means that the conversion system is made available across the Web; hence any author using MOT (which is also a Web-based system) can then access the service to convert their materials for use in WHURLE. Thus the conversion program no longer needs to be on every author's computer. On the other hand, *generalisation* means that by using open, Web-based standards, this opens up the conversion system. Other AEHS can submit input to the conversion Web service using the service definition. Other systems can also be added to the output. By implementing conversion as a Web service, the aim is to provide a middleware that offers an open API and hence allow system designers to create a conversion extension, which will then give access to many other AEHS, either as a delivery mechanism or as sources of content.

5.2.5 Conclusion

In this case study, an extension to the previous work in creating a new authoring paradigm, 'create once, use often', has been presented. The move towards this from a 'create once, use once' approach to AEHS authoring is an attempt to address the issue of authoring complexity. AEHS have yet to break into the mainstream educational culture, and this may in part be due to the cost involved in authoring for these systems (chapter 2). In allowing AEHS to exchange their learning materials, the initial cost of authoring is not removed completely but it can be ensured that authors only ever have to go through this process once. Interoperable AEHS will have future-proof access to these materials and allow for a more flexible delivery. This increases the choice of e-learning systems for the teacher and learner and increases the potential for personalisation and adaptive learning to be used more widely.

5.3 CASE STUDY 2[68]: Interoperability of Rich Content[69]

5.3.1 Introduction

One of the interesting problems that has attracted the attention of both industry and academia in current e-Learning environments is the interoperability of learning content in Learning Management Systems (LMS). A number of standards (chapter 2, section 2.7) have defined a Run-Time environment in order to allow an LMS to deal with learning objects i.e. to launch, track and communicate with them. Such standards achieved high performance when dealing with simple learning objects such as a text file, picture or a simple animation, but when dealing with more complicated content such as rich intelligent or adaptive content they show many limitations. Therefore, much research is ongoing in order to find a better solution and overcome such limitations. In this case study a closer look was taken at the issue of interoperability of intelligent learning objects which followed one of the non-standardised research efforts. Two different existing adaptive LMS (developed by different research groups) integrated an independently developed learning activity and provided it to its local learners. Prior to the integration

---

[68] Work presented in this case study involved collaborating with two other research groups from the University of Pittsburgh (USA) and Trinity College (Ireland). The research work presented in this case study was sponsored by the KESP project together with the National Science Foundation of the Republic of Ireland.

[69] The research presented here was published in the following papers (can be retrieved from http://www.cs.nott.ac.uk/~mzm):

Meccawy, M., Brusilovsky, P., Ashman, H., Yudelson, M. and Scherbinina, O. Integrating Interactive Learning Content into an Adaptive E-Learning System: Lessons Learned. In: Proc. World Conference on E-Learning in Corporate, Government, Health Care, & Higher Education (e-Learn 2007). Quebec City, Canada Oct. 15-19. 2007.

Rey López, M., Brusilovsky, P., Meccawy, M., Díaz Redondo, R., Fernández Vilas, A., and Ashman, H, Resolving the Problem of Intelligent Learning Content in Learning Management Systems. *International Journal on E-Learning*.2007. 7 (3).

process of a learning activity, some questions were raised and needed to be addressed and agreed upon between the LMS and the activity providers. Those questions were: What kind of information is exchanged in each step? Which components exchange the information? Who initiates the exchange? Which protocol and format is used during the exchange process? Who ends the learning activity and how would this termination state be acknowledged by the different components?

Before conducting the investigation presented in this case study, a number of research proposals and research standards such as ADL SCORM RTE[70] and IMS Shareable State Persistence[71] were examined. The survey showed that they provided different answers to these questions and have supported different sets of exchanges, each with their advantages and drawbacks. However, as mentioned earlier in this section, the shortcomings of those e-learning standards were clear (Rey López *et al.* 2007).

For this specific case study, it was decided to use ADAPT² (Brusilovsky 2005) and APeLS (Conlan 2002) as the LMS or learning portals for the implementation and integration process. This decision was made for a number of reasons. Firstly, they have been used to teach real courses to real students for several years. Secondly, they are ongoing research projects that have a lot of potential for future development and improvements and they were developed by pioneers in the field. Thirdly, the research was conducted collaboratively with their development groups.

[70] http://www.adlnet.gov/scorm/index.aspx

[71] http://www.imsglobal.org/ssp/index.html

The integration process was aiming to allow those systems to communicate with an independently-developed learning activity: *Boolean Queries*[72] or BQ. The Boolean Queries activity is an application which aims to provide a student with a firm understanding of the logical operators (AND, OR and AND_NOT). It uses the theme of a database of books where a student can search for a specific book using different parameters such as its ID, title, author's name, year of publication or publisher. What he/she will get as a result is a visual representation marking the conditions that were matched or mismatched in a distinctive manner (i.e. using different colours). This activity can be used as a part of a variety of existing Computer Science or Engineering courses such as Introduction to Databases, Digital Logic or Computer Organization. The integration of the external learning activity into the two adaptive systems was achieved using ADAPT^2 original protocols, while the activity itself had to be modified to use these protocols.

5.3.2 ADAPT^2 and APeLS

- **ADAPT^2** (Chapter 2, section 2.9.1)(Brusilovsky 2005) is an extension of the Knowledge Tree architecture (Brusilovsky 2004b), which is a distributed architecture for adaptive e-learning based on reusable intelligent activities. KnowledgeTree has been used since 2002 to support the learning of several courses at the University of Pittsburgh. ADAPT^2 is presented in the following figure (5.6):

[72] Developed originally by Olena Scherbinina

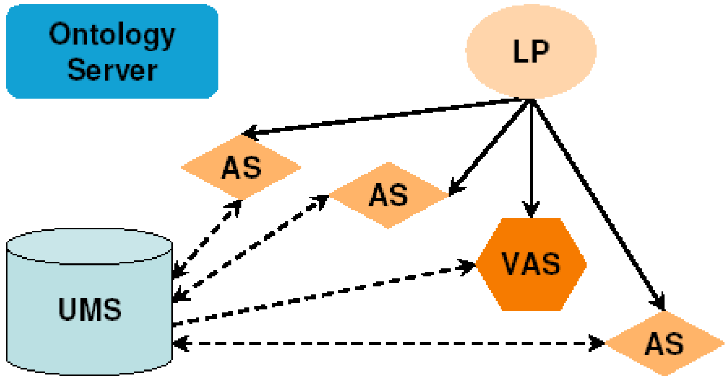

Figure 5.6: ADAPT^2 Architecture, taken from (Rey López *et al.* 2007).

Four main components construct the ADAPT^2 framework: the (1) **Learning Portal** (LP) is similar to modern LMS but the content is not embedded. It organizes the learning material and provides student and teachers with the facilities or features necessary for participating in the learning process. The (2) **User Model Server** (UMS) stores information about each user and the user activities to be used to infer the user's level of knowledge and learning characteristics. The (3) **Ontology Server** stores the ontological structures of the domain models. A Domain Ontology is a network of domain model concepts (topics, knowledge elements) which defines elements and the semantic relationships between them. Its purpose is to resolve the possible conflicts in the domain models of specific applications. In addition, it provides the platform for exchanging high-level information about students' knowledge, calculated by different user model servers (from different applications which might have their local user modelling servers). The (4) **Application Server or Activity Server** (AS) implements one or more kind of learning activities which might be adaptive or non-adaptive. It typically implements a piece of rich interactive or intelligent content that will engage the user on different series (steps) of activities over a period of time. Finally, the (5) **Value-added Service** (VAS) adds some additional interactive value to the raw content provided by Application Servers - for example, adaptive navigation support.

A teacher uses the Learning Portal to develop a course using one portal and many activity servers and services. While the student uses this portal to surf through this course and learn, he/she will be interacting with many learning activities served by different activity servers. In this distributed architecture, the UMS will provide the basis for performance monitoring and adaptivity.

- **APeLS** (Chapter 2, section 2.9.2) (Conlan 2002), was developed at Trinity College Dublin, and is currently being used to teach personalised online courses on SQL to final year undergraduate students in the college; it has also been used in a number of other projects. The following figure shows its architecture:

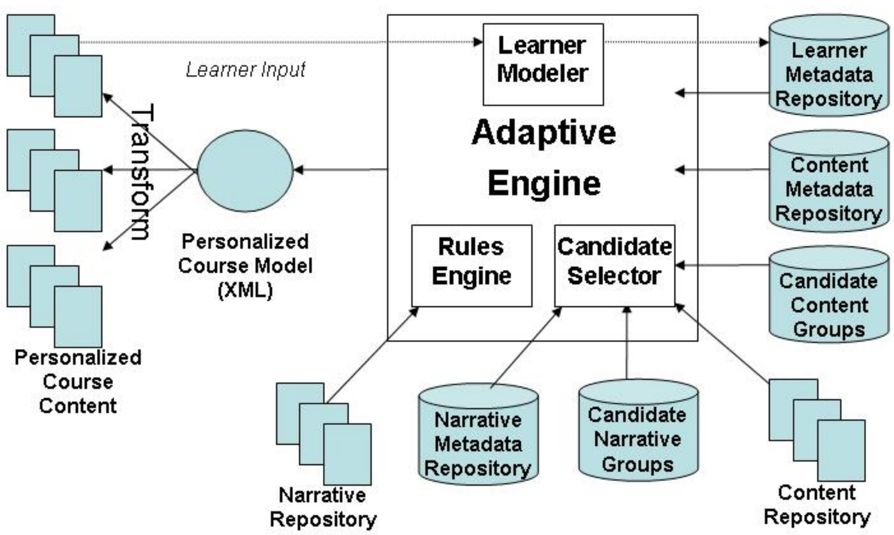

Figure 5.7: APeLS Architecture, taken from (Conlan 2004a).

APeLS was developed as a service to deliver personalised educational courses based on a multi-model, metadata-driven approach. The architecture of APeLS consists of three models: the learner, narrative and content models. The APeLS also has a number of metadata and information repositories. This metadata used in is an extension of the IEEE LOM[73] and ADL SCORM[74] (chapter 2, section 2.7.1). These repositories are described below::

• **Learner Metadata Repository:** stores all of the metadata representing the individual learners in the system.

• **Content Metadata Repository:** stores all of the metadata conforming to the Content Model corresponding to each piece of learning content.

• **Content Repository:** stores all of the page-lets referred to by the Content Model Repository.

• **Narrative Metadata Repository:** stores the metadata records that describe the learning objectives and pedagogical approach for each narrative in the narrative repository.

• **Narrative Repository:** stores all of the narratives used to construct personalized courses.

• **Candidate Content groups:** reference the groups in the Content Metadata repository that fulfils the same learning goal.

• **Candidate Narrative groups**: contain groups of narratives that encapsulate the same knowledge, but employ different pedagogical approaches to structuring the content.

[73] http://ltsc.ieee.org/wg12/
[74] http://www.adlnet.gov/

At the heart of APeLS lies the Adaptive Engine (AE); the models are taken at runtime and passed to this engine. From the AE, a personalised course is then delivered for an individual user. This uses the Java Expert System Shell (JESS) with customized functions as the basis of its rules engine. Taking a narrative and a learner model as its inputs, the AE produces a model for a personalised course.

Two important features of the architecture address the issues of interoperability and content reusability: 1) the adaptive courseware is not offered as content but rather as a service that can be delivered through a portal (APeLS provides a service interface where the adaptive course can be delivered). In addition, the interface provides a separate API to pass administrative and learner performance information to a portal or a LMS. 2) The adaptive service is driven by separate models (as described above) of content, narrative and learner model.

5.3.3 Integrating Boolean Query with ADAPT^2 and APeLS

As mentioned in the previous section, when an attempt is made to launch and integrate a learning activity with a foreign independently-developed adaptive system, some issues need to be agreed upon in advance. Answers for the following questions needed to be provided.

1. How is the user being authenticated?
2. How is the information about the user passed to intelligent content?
3. How can the trace of the user work be stored?
4. How can the end of the work with the content be registered?
5. How can the student model accumulated by intelligent content be shared with other components?

When integrating the (Boolean Query) learning activity with ADAPT² and APeLS, it was decided to use the ADAPT² protocols, which are original HTTP protocols, as well as its User Modelling Service (UMS). Hence the LMS (ADAPT² and APeLS portals) had to provide three compulsory parameters when launching the learning activity by calling its dedicated URL. Those parameters are the: user name, session ID and student's group.

The Boolean activity itself had to be modified to accept those parameters and to be initiated when the activity's URL is fired. The activity appended those parameters (in addition to others) to this UMS's URL (in order to communicate with the UMS), which passed them to the UMS. In the UMS, the student and his/her knowledge level was registered for later viewing of the user's trace. The user modelling implementation at the UMS side was the responsibility of the LMS provider (ADAPT² and APeLS developers groups). When the student finished his/her work by closing the activities window or logging out of ADAPT² or pressing the *Done* button in APeLS, the portal could then request the UMS for his/her profile. Figure (5.8) demonstrates the mechanism of this process:

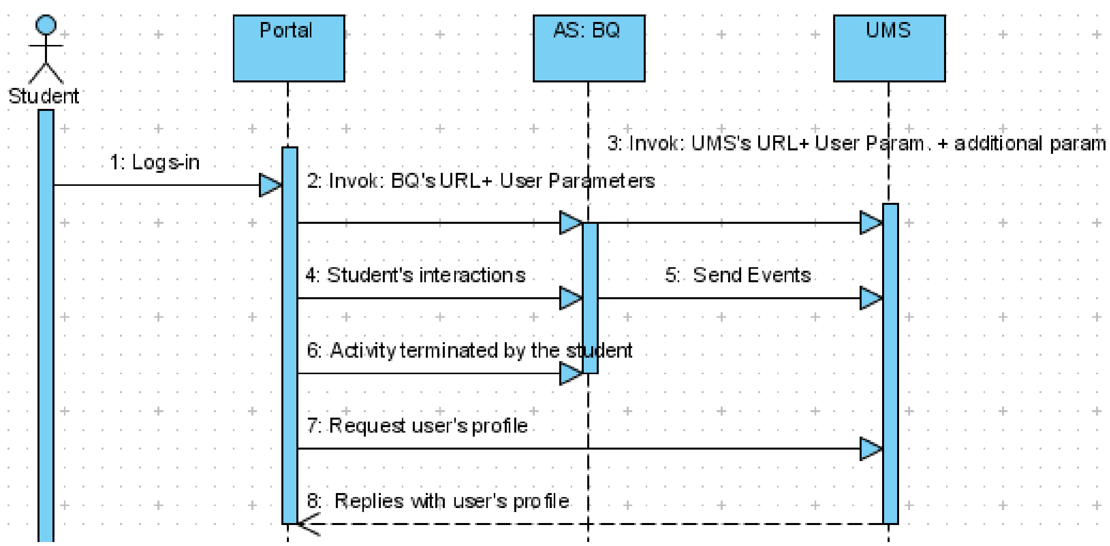

Figure 5.8: The process of integrating an adaptive portal and the learning activity (BQ).

while figures (5.9) and (5.10) shows the activity after it has been launched by the ADAPT² and APeLS portals respectively:

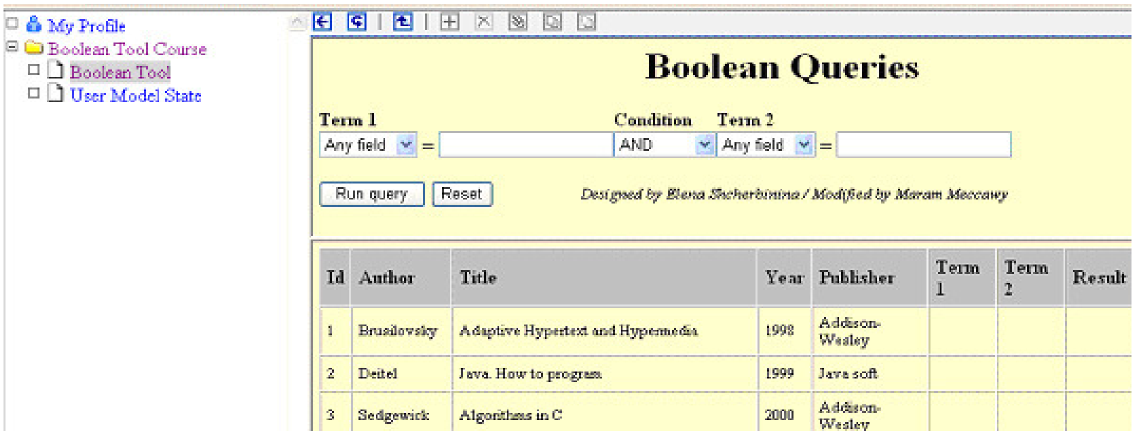

Figure 5.9: The learning activity as it appears within ADAPT² portal.

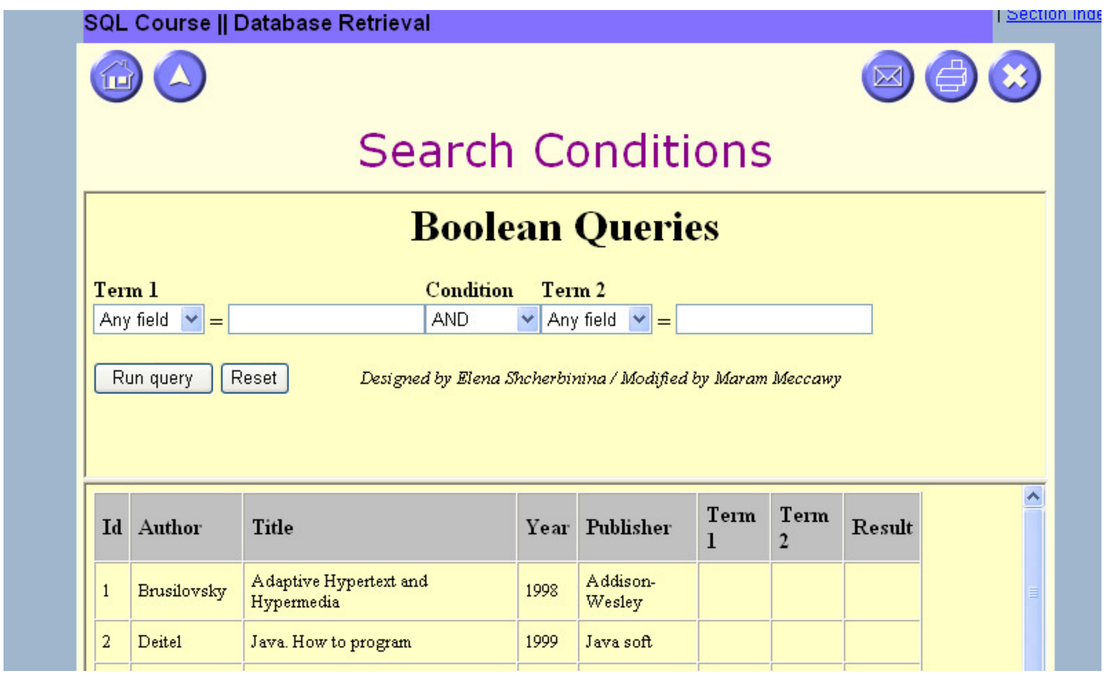

Figure 5.10: The learning activity as it appears within APeLS portal

5.3.4 User Trace

Being able to access the user trace has many benefits for all the stakeholders in the e-Learning process. The original system that produced this data requires access to it to provide continuity and proper adaptation. The LMS needs it to recognise the completion of educational objectives and to organise sequencing of learning content. Other types of intelligent content (used by the same student) have to use an updated state of the student goals and knowledge observed by an activity in order to provide adequate adaptation. The teacher may wish to monitor the student's work with an activity to recognise problems and to assist him/her. The students may seek to observe the growth of their knowledge over the use of various learning activities. Overall, the ability to access and use this information is one of the keys to the success of intelligent content in modern e-learning. The research work therefore provided teachers or course administrators with a special *teacher view* for reviewing and replaying the user's activities. This will give them a picture of *what* has been done (searched for in this case), *when* those actions have taken place and *who* has performed them (i.e. student's name). This feature can be important and extremely beneficial in cases where students are taking online exams, in order to answer students' complaints about their score and performance.

5.3.5 Lessons Learned

The research work presented here has shown practically that the interoperability of rich, intelligent or adaptive content is still a problem that needs to be resolved in order to allow different LMS to launch this type of learning materials, which has a new and more significant impact from a pedagogical point of view than has simple content. It is this type of courses or activities that is most likely to push e-learning forward in the near future.

A better definition is needed of a Run-Time environment which allows an LMS to deal with intelligent content (launch, track and communicate with it) and satisfies all its requirements.

In addition, the use of original or proprietary protocols such as those seen in ADAPT² is not a realistic, practical or feasible solution for real world settings since a lot of human communications were involved throughout the integration process between both sides i.e. between content developer/provider and any LMS provider that wishes to use it. This integration should therefore be made as autonomous as possible by using other technologies which provide this capability, such as the emerging Web services' technologies. Some research groups have taken this step forward as in MEDEA (Trella *et al.* 2005). Finally, it is important for all partners involved in the e-learning process to have the ability to access the student's user model or view the student's trace that has been recoded during his/her interaction with the system. A modern LMS should be able either to store this information about the student locally and have it available for other components, or to have the ability to request a certain user's profile and access this information that is stored at (or by) another component such as the UMS in ADAPT².

5.4 Summary

After analysing the issue of interoperability in chapter 2 of the literature review, this chapter focused on investigating interoperability of adaptive learning content through empirical research. The first case study extended previous work in creating a new authoring paradigm, 'create once, use often' that followed the conversion approach. However, it has suggested a Web service approach would make it more flexible and accessible, leading to an increase in the potential for personalisation and adaptive learning.

In addition, the second case study investigated the issue of exchanging rich content verses simple content. The latter is straightforward and has been addressed by a number of e-learning standards or initiatives such as SCORM but achieving this degree of interoperability or reusability for the former, is more complex. A number of steps were involved in the integration of a rich piece of content or in a learning activity and hence it could not simply be packaged and copied from one system to the other. Some other research efforts are therefore needed. The work described here is one which worked closely with two existing adaptive learning systems in order to achieve the reusability goal of adaptive, interactive or rich e-learning content.

Building on the experiences learnt from the case studies presented in this chapter, a new architecture for adaptive systems which has interoperability as one of it core requirements has emerged. The next chapter presents this new architecture.

CHAPTER 6

"Everything should be made as simple as possible, but not one bit simpler"
Albert Einstein (1879 - 1955)

FROM WHURLE TO WHURLE 2.0: A SERVICE ORIENTED ARCHITECTURAL DESIGN

6.1 Overview

The previous chapters introduced and investigated the current challenges facing Adaptive Educational Hypermedia Systems (AEHS), which are believed to result in their absence from the e-learning community. Among those challenges are issues of limited interoperability and reusability of content, user information and system components. While different approaches have been proposed to solve these issues, addressing the problem at a broad architectural level is seen as a well-justified approach. Re-engineering the architecture of e-learning systems in general using modern technologies such as Web services (chapter 4, section 4.5), is argued to be a valid method to address its challenges, since only with effective engineering and structures could there be sustainable solutions for real world challenges.

The key contribution of this thesis lies in the specification of a new architecture, which is named WHURLE 2.0 that will eventually advance the cause for adaptive e-learning systems. This architecture is one of a service-oriented nature which is designed to provide not only an adaptive e-learning system but also to present an extended rich and interactive e-learning environment that enables collaborative learning to co-exist with adaptation.

WHURLE (Brailsford 2002b), is one of the pioneering adaptive systems of the beginning of the third generation. WHURLE 2.0 has retained the adaptation concepts from the former system, but nevertheless, has taken those concepts to another level, whereby they become more independent physically as well as conceptually.

The focus of this chapter is on the conceptual design aspects of the architecture while details about its development are presented in the following chapter (7).

6.2 The WHURLE Framework

WHURLE: Web-based Hierarchical Universal Reactive Learning Environment (Brailsford *et al.* 2001; Brailsford 2002b, a; Zakaria *et al.* 2002) is an online learning system designed to provide a discipline-independent framework, which is pedagogically flexible and is capable of providing adaptation. The adaptive components of the WHURLE architecture depend on one of Nelson's original visions of hypertext, which is transclusion or conditional transclusion in the case of WHURLE. Transclusion consists of:

> *"dynamic inclusion of arbitrary components of one document inside the other"* (Nelson [6] as cited in (Brailsford *et al.* 2001).

WHURLE implemented a simplified model of transclusion where the content is adapted to the needs of the user to create *virtual documents* that only the learner can view. The conditions respected in this context are the information stored in the user profile, which allows the creations of such documents dynamically to match the needs or requirements of the learners. WHURLE was previously used to teach a Biodiversity module at the University of Nottingham.

6.2.1 WHURLE's Architecture

WHURLE stores educational content in separate files called *chunks*. Lessons are created, by educators, to provide a default pathway in the *Lesson Plan* through the available chunks. In the lesson plan, a teacher defines a hierarchy of pages (each consisting of one or more chunks) that can then be navigated by learners. This default pathway is adapted to the needs of individual learners by reference to their *user profile*. WHURLE also provides a robust linking system, and automatically-generated navigation around the lessons.

The WHURLE framework was designed to be modular, separating the functions of adaptation and rendering. During parse time the system builds a node-tree from the lesson plan. This is then processed by the *adaptation filter*, a XSLT style-sheet that would implement the lesson plan against the user profile. The result produced by the adaptation filter would be then passed to the *adaptation engine*, which is also an XSLT style-sheet that renders the final output to be presented to the learner. In creating this output, the engine overlays the navigational system on the top of the learning content (chunks) in addition to a user's interface that is described by the *skin*. The skin is an XML file which defines the appearance and any associated learning tools, links or resources. Finally the output document (the lesson) is a dynamic HTML document.

The concept of XML pipeline[75] is fundamental to the WHURLE architecture and it is a:

---

[75] Pipelining is technique that allows multiple requests to be sent out to a single socket without having to wait for their responses.

"*series of events generated at parse-time, that follow through a predefined sequence of filters or processes*"(Zakaria 2003).

It resembles the Unix pipeline which uses the outputs of one program as an input for another. Hence all the outputs of the XSLT transformation of the XML documents (described above) are passed to this pipeline and then to the display engine, all which occurs transparently where the learner receives only the adaptive virtual document.

A conceptual description of the WHURLE architecture and how its components collaborate to produce the adaptation effects is presented in figure (6.1) while figure (6.2) shows a snapshot of one of WHURLE's end documents as it appears to a learner, available in figure (6.3).

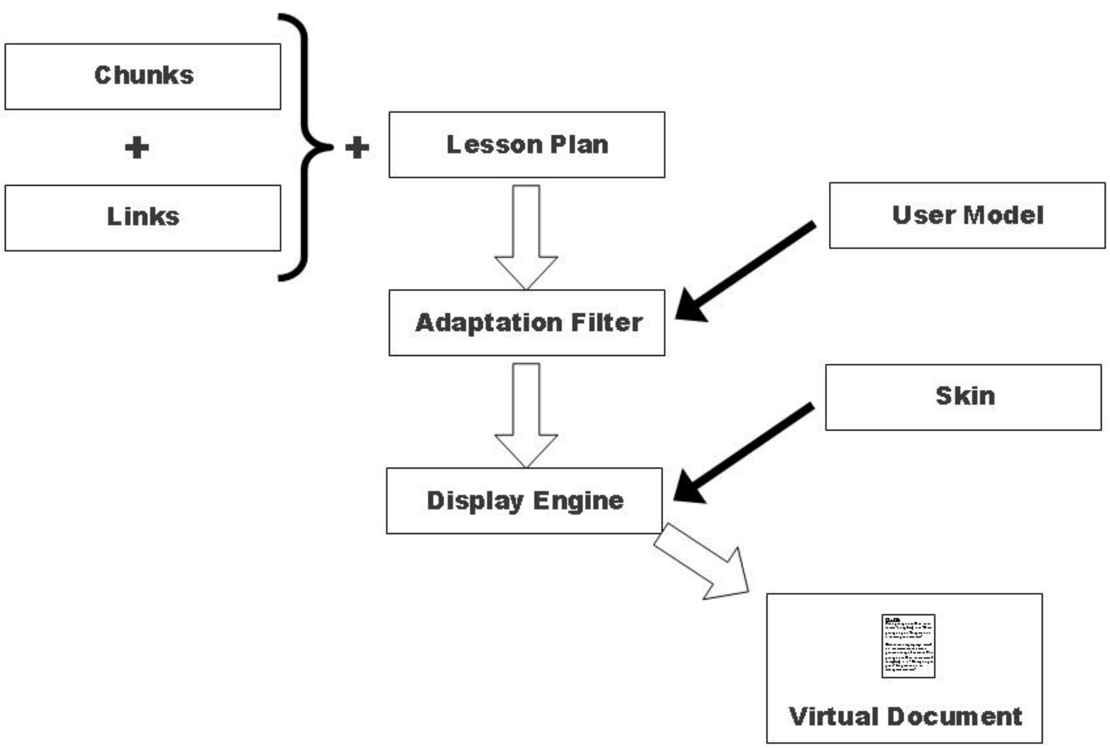

Figure 6.1: The architecture of WHURLE, taken from (Brailsford 2002b).

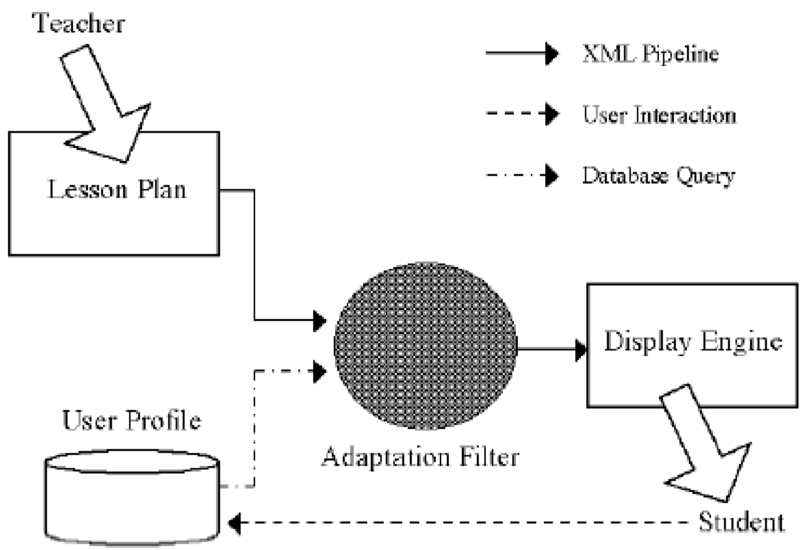

Figure 6.2: "The adaptation filter in WHURLE. A teacher creates a lesson plan that defines all possible content of a lesson. This is filtered according to the user model that makes use of information stored in the user profile. The output of the filter is rendered by the display engine and student interactions update the user profile", taken from (Zakaria 2003).

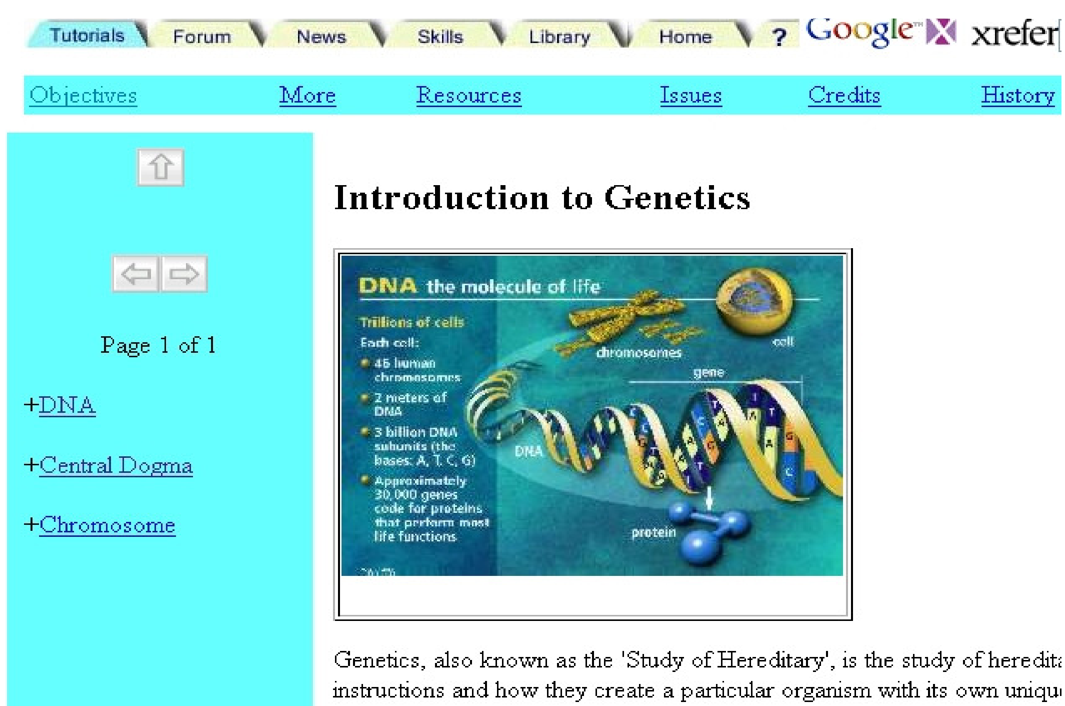

Figure 6.3: A snapshot of a WHURLE's lesson, taken from (Stewart 2003).

6.2.2 The Limitations of WHURLE:

The WHURLE framework aimed to be modular to a large extent to allow easy modification of its architectural components. It was designed to be flexible and not tied to any specific user interface or user model. However, different implementations of WHURLE had their own issues and generally presented closed systems where the interdependencies between different components were significant and as a result limited the modularity as intended by the original framework. Nevertheless, the WHURLE framework was designed to take into account the technologies and demands of its time. Achieving interoperability of content or user profiles between different adaptive systems was not one of the design's core objectives. In order for WHURLE to use content developed in other systems the external content or user profile needed to be converted to WHURLE's own formats, as described in the MOT to WHURLE conversion (also see chapter 5, section 5.2).

WHURLE did not have any built-in collaborative learning tools but it provided hooks that allow the inclusion of external social tools (such as Forums) by specifying their URLs. Different implementations or versions had therefore enabled the use of a variety of tools such as Chat, Forum or News. However, the ability to integrate WHURLE (with its adaptation capabilities) into a popular LMS, which has a set of social tools in addition to other desired functionalities, would have been a very challenging task. This is due to different components being interconnected together including those which are responsible for preparing and presenting the output using the WHURLE's interface.

To conclude, WHURLE, along with the majority of AEHS, suffered from a number of issues that are not directly related to its useful adaptation functionality, but rather to its architecture and implementations, which might have served their purposes at design time but became limited in the context of the current demands for e-learning. WHURLE 2.0, on the other hand, was designed to address those limitations and serve those demands.

6.3 WHURLE 2.0 Specifications and Description[76]

6.3.1 The Architecture: An Introduction

In addition to the limitations of WHURLE identified in the previous section, the advances of Web 2.0 technologies have prompted a re-think of the WHURLE implementation design. In this new design, the system's components (i.e. user modelling component, adaptation filter, lesson plan and the learning resources) become a set of modular, autonomous, independently-developed services that can communicate with others by adhering to a specific set of protocols that have been previously agreed: Web services protocols, namely SOAP and WSDL. All the services share values of independency, interoperability and flexibility.

In WHURLE 2.0, the learner is presented with a workplace rather than a single document and hence the system's *Portal* or delivery service provides the single login point to this learning space or environment. When a user logs into this portal, it acts as a

---

[76] Parts of this section were published as described below:
Meccawy, M., Blanchfield, P., Ashman, H., Brailsford, T., and Moore, A. WHURLE 2.0: Adaptive Learning Meets Web 2.0. In: Proc. EC-TEL 08: 3rd European Conference on Technology-Enhanced Learning Maastricht, The Netherlands: Springer Verlag, September 16-19. 2008.

Meccawy, M. WHURLE 2.0: An Adaptive Web Service e-Learning Environment for the Web. In: Proc. Saudi Innovation Conference (SIC 2007). Newcastle upon Tyne, UK May 11-12. 2007.

client and communicates in the background with other system components (services) to perform the required task.

WHURLE 2.0 is composed of six main components that are explained in detail in the following sections. These are five independent Web services that collaborate with one another to tailor a unique view of the learning content for a given learner in addition to a delivery service where the learner views this adaptive content. Those Web services are the: *Aggregation Service* (AGS), *User Modelling Service* (UMS), *Lesson Plan Service* (LPS), *Adaptation Filter Service* (AFS) and the *Chunk Management Service* (CMS), while the delivery service for the new learning environment was chosen to be a Learning Management System or an LMS (for example Moodle). Figure (6.4) below provides a conceptual description of WHURLE 2.0's architecture; this is followed by a description of each of those components which together compose the WHURLE 2.0 architecture.

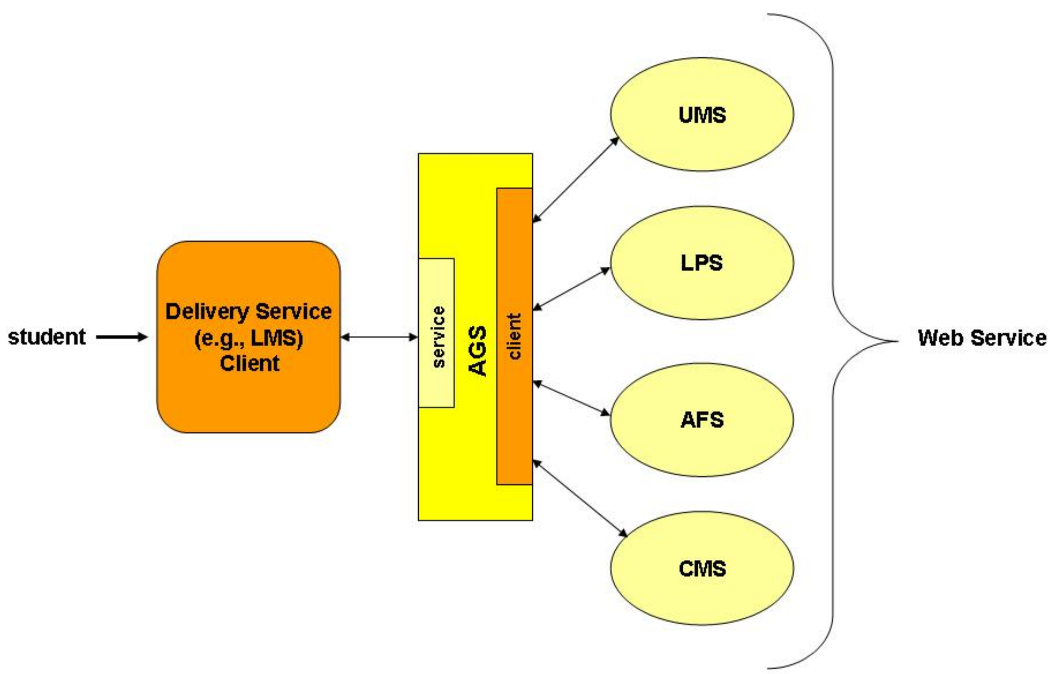

Figure 6.4: WHURLE 2.0 Conceptual Design

6.3.2 Aggregation Service (AGS)

Since WHURLE 2.0's architecture is a SOA that describes a distributed environment, where many different services are communicating with one another constantly, therefore it is important to provide an *Aggregation* Service (AGS), which deals with different requests and responses and which manages the intra-system's communications. The AGS is invoked as a service by a client (in the portal or LMS), which calls this service and passes the required parameters. Then the AGS itself becomes a client which invokes other services before returning the final result to the initial client call.

AGS can be viewed as the central point that aggregates different components of a distributed and service-oriented architecture; it glues those parts together to perform the main task. Comparing WHURLE 2.0 with the former WHURLE, the AGS has been introduced to replace the XML pipeline operation where different components of the original WHURLE had to pass their results in a monolithic environment. In addition, the AGS provides an interface between the delivery service, which is the LMS, and the external adaptation services.

The AGS (as a service) is invoked by the portal (or the LMS) and then it will make its first communication (as a client) to call the User Modelling Service (UMS) for registering the user, retrieving the profile of an existing user or updating this user's profile. The UMS is discussed in the following section.

6.3.3 User Modelling Service (UMS)

The User Model Service (UMS) plays a number of roles in WHURLE 2.0: 1) it is the service where the modelling of the user takes place; 2) it is where users' profiles are stored and retrieved; and 3) it is responsible for updating those profiles. The UMS communicates with the AGS to register new students in its database, to respond to a request from AGS by providing the user's profile at the beginning of the adaptation process, or to accept updates of the students' profiles.

Having a distributed user modelling service helps the end-users to have the flexibility of using different LMS that provide different courses. Moreover, if they are given access to their profiles, then they can stay in control of those profiles and modify them if needed. In the implemented version of WHURLE 2.0, the user is modelled according to his/her knowledge level using a stereotype modelling technique, where there are essentially three levels: Beginner, Intermediate and Advanced. The student is classified into one of those three levels and this information is stored in the UMS database. Chapter 7 provides further details about how the modelling was achieved (sections 7.2.4).

6.3.4 Lesson Plan Service (LPS)

The Lesson Plan Service (LPS) is where the lesson default pathway is specified. The teacher states which concepts (or chunks) should be taught to each type of user and in which order each lesson plan is presented, a procedure which is similar to that of former WHURLE described in section 6.2. The difference with WHURLE 2.0 is that the Lesson Plan becomes a service, which stores a variety of lesson plans for different courses; hence a database that keeps records of those lesson plans was needed. When the Service is invoked by the AGS's call (carries the lesson's name as parameter), the LPS responds by locating the matching lesson plan for that lesson. The AGS then passes the result to the Adaptation Filter Service, which is explained in the next section.

6.3.5 Adaptation Filter Service (AFS)

The Adaptive Filtering Service (AFS) implements the user model against the lesson plan; it is the engine which triggers the adaptation effects and decides on the content to present to a given learner. It receives a request from the AGS for both a user's profile and a lesson plan. The AFS responds by applying its rules, which results in the tailored filtered lesson being specified in a list of chunks. Those chunks will then be pulled out from the *Chunk Management* Service (CMS) as illustrated in the next section.

6.3.6 Chunk Management Service (CMS)

Following the original WHURLE's design, all the chunks in any one instance of WHURLE (i.e. a single installation on the server) are referred to as the melange (Brailsford 2002b). In the melange, there was no central database that acts as an interface for any possible external communication with those local chunks. This fact would have been a key barrier for interoperability of learning content (chunks) with any other system. In WHURLE 2.0, therefore, the Chunk Management Service (CMS) is the service responsible not only for serving all the chunks (learning content) to the requesting client, but in addition, the CMS stores knowledge about the subject's concepts to be taught, the relationships between them, their location (stored locally or externally) and other information such as the concept's level of difficulty. Since WHURLE 2.0 is designed to address interoperability issues and integrates a wide variety of resource-compliant learning activities, having a service that stores content information was essential to provide some meta-data for interoperability purposes.

When the CMS receives the AGS call with the list produced by the AFS, it locates the named chunks, extracts their content and sends it back to the AGS that in turn passes the selected lesson, which is composed of one or more chunks, to the LMS (portal), where the student can access it.

6.3.7 A Learning Management System: A Portal and Delivery Service

As mentioned in chapter 3 (section 3.3), open source modern LMS provide a number of tools to assist different requirements of an online learning process from content creation to learners' activities and communications, which comply, to some extent, with the characteristics of the new Web 2.0 technologies (Millard and Essex 2007). This learning tool or system is therefore chosen to be the platform where adaptive learning takes place. Hence the capabilities and functionalities of adaptive and personalised learning are combined with the values of sharing, peering, communicating and collaborating promoted through Web 2.0 tools provided by a modern LMS.

While designing the environment, the objective has been to make the integration of this adaptation layer transparent to the learner and for the implementation process to require minimal modification to the LMS software. Because of this, the layer is added externally and the communication between this layer and any database table, tool or activity in the LMS is established using agreed protocols of Web services. The argument here is that, instead of trying to gain a critical mass of support for a new adaptive system that satisfies e-learning 2.0 requirements, adaptation should be presented through an already-in-use LMS that covers the social aspects of online learning.

The LMS in WHURLE 2.0 replaced both the *Adaptive Engine* and *skin* components of original WHURLE (section 6.2), which were both essential and integral parts for preparing the output presentation for the learner and defining the system's user interface.

6.4 Discussion

Re-designing the WHURLE's architecture as a collection of independent Web services allows it to be integrated easily with other popular and widely used systems such as LMS without requiring in-depth knowledge of the latter's code. Web services are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Designing the adaptation components as individual Web services had many advantages. Each of those services could be upgraded, maintained or replaced individually without affecting the delivery systems (LMS) or any other service, which allows for greater flexibility and maintenance of the learning environment.

As mentioned earlier in this thesis Interoperability is needed on three levels: a technical level, which is achieved through Web services; XML, a structural level, achieved by re-designing the architecture and decomposing its components into independent services; and a semantic level which has been dealt with partially in CMS and UMS, though these still require further investigation.

Moving from the standalone to a service-oriented architectural design serves many purposes. It permits the reusability of content and interoperability of each system component; hence this architecture enables easy integration and maintenance of each individual service without affecting the other services. Moreover, an independent user modelling service helps to provide portable user profiles and therefore has the flexibility of using different LMS that offer different courses. In addition, the architecture enables the integration of new services, with only a single connection needed between this new service and the architecture via the AGS. The architecture of WHURLE 2.0 also made it possible for developers to address other issues affecting the majority of AEHS. Such issues include these AEHS being prototypic systems with experimental interfaces and limited social, collaborative or interactive tools, seen to be of critical importance in the Web 2.0 generation of application and users. The service-oriented approach therefore allows the integration of the adaptation services with an external independently-developed "off the shelf" popular LMS. This saves time and effort as well as providing both the teacher and the students with an adaptive system that is sufficiently robust, easy to use and equipped with different learning activities.

Hence, instead of re-inventing the wheel by building new adaptive systems and trying to convince people to leave their already-in-use LMS to gain the adaptation effects, it is rather more rational and logical to push the adaptation into those systems.

One possibility for a potential problem in this architecture is - if those services are located on different servers, then what happens if one or more of those servers experienced some technical problems that caused them to fail? This could mean that the whole system is no longer available to produce the adaptive content (created collaboratively by the different services). However, one way to deal with this problem is for the service provider to have substitute services located on another server, which would act as a backup in case the original server is out of service.

6.5 WHURLE 2.0 and WHURLE: A Design Comparison

The following table 6.1 provides a summarised comparison that shows the most significant differences between the two distinct adaptive systems WHURLE and WHURLE 2.0:

Category	WHURLE	WHURLE 2.0
Architecture	Monolithic	Distributed SOA
User Interface	Produced by the adaptation engine, skin and auto-navigation	LMS
Navigation	Auto-navigation and linkbase	Delegated to the LMS (could also be implementation-dependent if a new service is created for navigation)
Communication between components	XML pipeline	AGS
Chunks Meta-data Description	implementation dependent	CMS
User Model Meta-data Description	-	XML database
Social and collaborative Learning Tools	Provided hooks for adding external tools, implementation dependent: (Forums, Chat)	Tools provided by the LMS (or any other delivery system). Tools provided through the implementation provided in this thesis included: Chat, Forums, A Wiki and a Blog

Table 6.1: WHURLE and WHURLE 2.0: A Comparison of their architectures.

6.6 Summary

The limited uses of adaptive e-learning systems were believed to be due to their architectures failing to answer the current modern requirements of the e-learning community. Therefore by redesigning their architectures, a sustainable solution is provided which will help in promoting the use of those systems outside the laboratories.

WHURLE was taken as an example of a successful adaptive system, which itself has issues that limit its usage. By re-engineering WHURLE’s architecture, transforming it from a monolithic adaptive Web-based educational system into a service- oriented integrated e-learning environment, while preserving its own adaptation concepts, a new architecture has emerged. This new architecture, WHURLE 2.0, has the ability to address the immediate and future demands of modern e-learning. WHURLE 2.0 uses Web services to insure interoperability and it allows easy integration with any delivery system as well as between the different adaptation services or components.

The development and implementation of this architecture is presented in the next chapter.

CHAPTER 7

"The only thing that overcomes hard luck is hard work"

Harry Golden (1902 - 1981)

IMPLEMENTATION

7.1 Overview[77]

The development and implementation process aims at providing an adaptation layer that will be integrated externally into a free source Learning Management System (LMS). The well-known LMS, Moodle (chapter 3, section 3.3.1), is chosen primarily to be the delivery service for hosting the adaptation in this integrated Web-based learning environment.

Chapter 7 builds on the previous chapter that has explained the conceptual design of the WHURLE 2.0 framework. This framework is composed of five Web services, which are: the User Modelling service (UMS), the Lesson Plan Service (LPS), the Adaptation Filter Service (AFS) and the Chunk Management Service (CMS) in addition to the delivery service which is the LMS Moodle in this implementation. The chapter describes in detail the implementation of the components of the system and shows the connections and interactions between the developed services themselves as well as between the services and the LMS Moodle.

[77] All the code files described in this chapter are provided within the attached CD (appendix-D).

7.2 Developing WHURLE 2.0

7.2.1 Introduction

The implementation of the Web services is achieved using PHP, XML and XSLT (chapter 4, section 4.2). In addition, it was one of the development objectives when integrating those services with the LMS Moodle, to use tools provided by Moodle where appropriate. These included Moodle's resources and activity modules as well as some of its database tables, making the integration easier for the developer and transparent to the learner.

Three clients are developed and installed in Moodle to allow for: the student's registration with the UMS (Client1); the adaptive content selection (Client2); and the updates of student profiles (Client3).

The adaptive learning process through WHURLE 2.0 starts by registering the user first to Moodle's end, and then to the external UMS, which is achieved through a Pre-test using Moodle's *Quiz* activity (chapter 3, section 3.3.1). When the student takes the Quiz and submits his/her answers, the first client is fired to perform the registration and to assign the appropriate knowledge level in the corresponding user profile.

After that, the student selects an adaptive lesson using Moodle's (built-in) *Lesson* activity; this selection activates the second client which will select the adaptive content and present it to this student. When the student wishes to log out the system (to the student this means logging out of Moodle), the third client is launched to update his/her profile. Alternatively, the user profile could be updated as a result of taking another quiz or quizzes; the only difference is the module or tool which will trigger the third client to call the AGS requiring the update function.

The implementation of each component is explained individually in the following sections, starting with Moodle's end of the environment, where the students' communication and interaction with WHURLE 2.0 takes place. While figure (6.4) in chapter 6 (section 6.3.1) has shown WHURLE 2.0's conceptual design, figure (7.1) illustrates the mechanism of how WHURLE 2.0 functions and highlights different arguments (parameters) that are passed in this process.

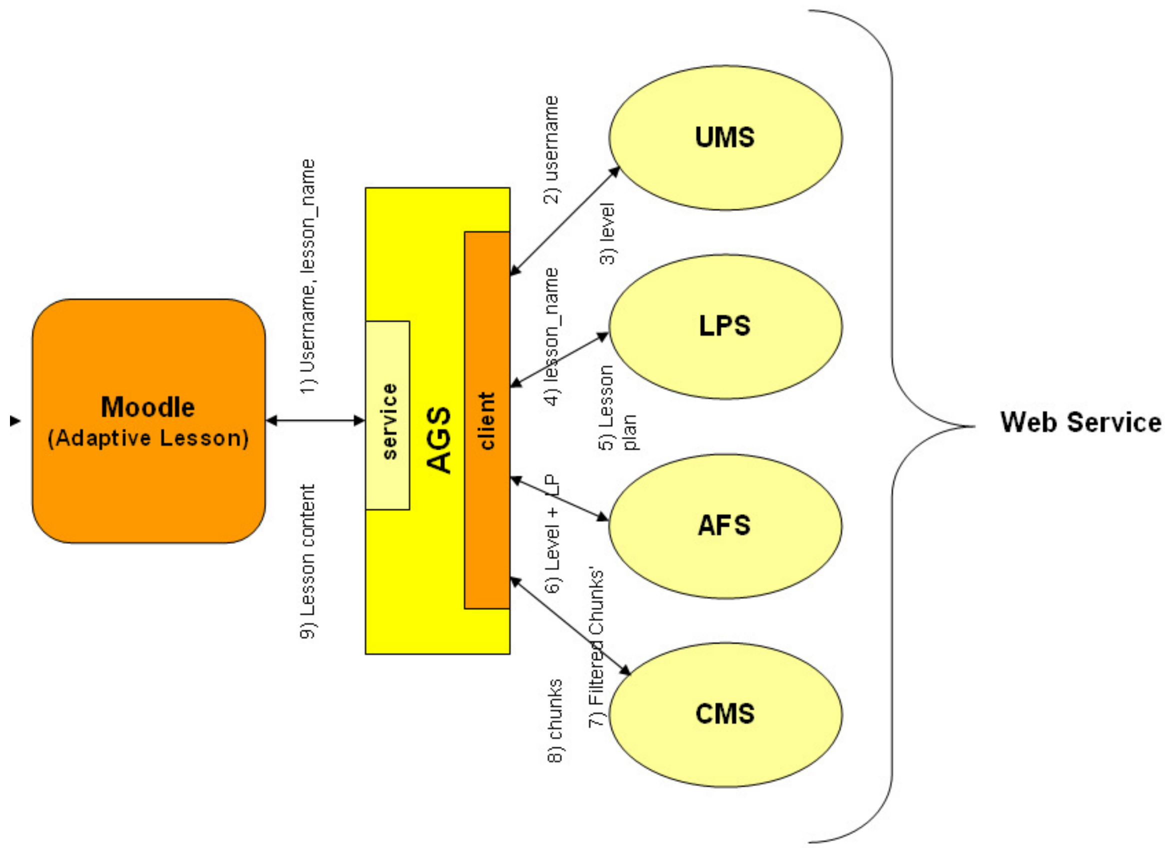

Figure 7.1: WHURLE 2.0: Integrating the services with Moodle

7.2.2 Moodle: Register, Login, Quiz Activity, Lesson Activity and Profile Updates

The LMS Moodle is the learner's end of the integrated WHURLE 2.0 adaptive and social learning environment. In order to use Moodle, students are required to register themselves with it. After they log into Moodle they need to be registered with another system, which is the backend of the adaptive system. This adaptive system is composed of five services including the UMS. The UMS (which is located outside Moodle) is where the students need to be registered and then assigned a level according to their knowledge in a particular subject. This is achieved (for example) by allowing the student to take a Pre-test on a specific topic (only once) using Moodle's Quiz activity; the quiz's grade is then used to detect the level of knowledge. Hence, when the student later selects the adaptive lesson, the level of knowledge would have been already decided and as a result the content presented will match his/her level.

This section continues to explain each step involved in this integration, starting from Moodle's end, as follows:

- **Quiz Activity:** the students are given quizzes that test their knowledge in a specific domain, area or topic. This is implemented using a scale from 1-9, where scoring 1-3 means that the student is a beginner, 4-6 an intermediate and 7-9 an advanced learner. The grade gets calculated by the Quiz and is stored in Moodle's MySQL database.

  When the first client (AGS_client1.php) is activated after the user has taken the quiz, it makes a call to the UMS through the AGS. During this call, the client carries a minimum of three parameters: the student's e-mail address, username and grade. When the client is fired, a call is made to the `AGS_register` function, which in turn calls the `Register_User_AND_Assign_Level` function in the UMS. This latter function compares the given grade and assigns to the user a certain level of knowledge (Beginner, Intermediate or Advanced) accordingly. First, the student's id (e-mail address) is searched for in the database to check if he/she already exists there; if not, then the student is registered to UMS with this id, username and level. Figure (7.2) shows a snapshot of Client1, while figure (7.3) describes Moodle's *mdl_quiz_grades* table:

```
$wsdl="http://bubba.cs.nott.ac.uk/AGS/AGS.php?wsdl";.

$client=new soapclient($wsdl, 'wsdl'); .

$param=array("id"=>$id,"username"=>$username,"grade"=>$GRADE);

$result=$client->call('AGS_Reg', $param);.
```

Figure 7.2: Client1: calls the AGS to register the student with the UMS and assign a level of knowledge according to the grade, which is in this case obtained from the student's Pre-Test (Quiz) Results.

id	quiz	userid	grade	timemodified
1	2	4	7.5	1204899233
3	2	8	7.5	1204823258
5	2	10	0.5	1205251957

Figure 7.3: One of Moodle's database tables: mdl_quiz_grade, where the student quiz grades are stored and then used as a parameter in client's call to the AGS

- **Lesson Activity**: in Moodle, the teacher prepares a lesson using the *Lesson* activity by specifying the name and other variables using the Lesson's built-in features. The only difference when using a Lesson activity for providing adaptive content is that instead of writing all the learning content in the pages using the

Lesson's built-in HTML-editor, the content page is left empty, to be filled with the adaptive content that is prepared later during runtime while the student is interacting with the system. When the student clicks on the Lesson's link as it appears in Moodle's Course webpage, the AGS's second client (AGS_client2.php) is activated calling the `AGS` function using the two parameters: username and lesson name. The AGS uses those parameters to invoke other services to prepare and return the content to the calling client, which uploads them to Moodle's database. A snapshot of Client2 is shown in figure (7.4):

```
$wsdl="http://bubba.cs.nott.ac.uk/AGS/AGS.php?wsdl";.
$client=new soapclient($wsdl, 'wsdl'); .

$param=array("username"=>$username, "coursename"=>$lessonname); .

$result=$client->call('AGS', $param);.

 $dbname = "mzm";.
 $dbuser = "";.
 $dbpass = "";.
 $dbserver = "localhost";.
 $tablename = "mdl_lesson_pages";.

$conn = mysql_connect($dbserver, $dbuser, $dbpass) or .
die('Error connecting to mysql');.
mysql_select_db($dbname);.
$sql_query ="UPDATE $tablename SET contents='" . .
mysql_real_escape_string($result)."' WHERE lessonid='$lesson->id'";.
$c_page = mysql_query($sql_query, $conn) or .
die('counter INSERT error: '.mysql_errno().', '.mysql_error());.
mysql_close($conn);.
```

Figure 7.4: Client2: calls AGS to prepare an adaptive lesson for the current student according to the level, and then pushes the result into the Lesson's table.

- **Profile Updates:** When the third client (AGS_client3.php) is fired, the user's profile gets updated. This could be done when the student logs out Moodle or alternatively after taking another quiz, for example a Post-Test, as has been chosen in this implementation. This client calls `the AGS_update` function and carries three parameters which are: id, username and grade The `AGS_update` function in turn calls the UMS's `Update_User_Level` function. The student's grade is compared with the current level and any updates will be made here. The next time the student logs into the system, therefore, the new level will be reflected when an adaptive Lesson is selected. Client3 is similar to Client1 except for the special AGS function that it calls, which in turn calls a different (third) function in UMS as will be explained further in section (7.2.4). In the experiment presented in chapter 8, no significant changes were reflected in the users' profiles due to the short period of this experiment.

7.2.2 Aggregation Service (AGS)

As mentioned previously in chapter 6, the Aggregation Service (AGS) is the heart of WHURLE 2.0. It is both a client and a service: a client that invokes other services and a service that is invoked by Moodle's (or any other LMS) clients (see 7.2.2). The AGS.php script implements all the AGS's functions and therefore contains all the calls to the other services. In addition, the AGS holds the temporary XML files that are created for storing data results from one service before passing it to another.

As mentioned in the previous section, the AGS has three main functions: `AGS_register`, `AGS` and `AGS_update`; those are explained further as follows:

- **AGS_register**: this function implements the function calls to the UMS for initial student registration to the latter's XML database. It receives three parameters: id, username and grade (see section 7.2.2, Quiz), and will then pass them to the UMS by calling its registration function.

- **AGS**: implements the function calls from the AGS to the UMS, LPS, AFS and CMS. This function receives the user's identification and the adaptive lesson's name from Moodle and passes them, as parameters attached to a service's call, to the UMS and LPS respectively. The results returned from those two services - the user's level (from the UMS) and the lesson plan of a specific lesson (from the LPS) - are then passed to the AFS, which applies (using these results) the adaptation rules and returns a list of required chunks to the AGS. After that, the AGS puts the results into an XML file (filteredChunksnames.xml) which is used to invoke the CMS. The CMS uses this list to select the chunks and extract the lesson's content before passing this content back to the AGS. The AGS then returns the newly-prepared adaptive content as a response to the initial client call received from Moodle. The client in Moodle is now responsible for populating Moodle's MySQL databases with the lesson content as mentioned in the previous section (7.2.2). This saves time since there is no need to create new tables.

  Since the client is activated to fill the tables responding to a student's selection and the content is prepared instantly during runtime, there are no synchronising issues between the tables in Moodle and the adaptive content. If any changes are made to the original chunks, this will be reflected immediately in Moodle's lesson as soon as the page is refreshed or when the lesson is next chosen.

- **AGS_update**: this function implements the calls to the UMS for a student profile update. It receives three parameters (see section 7.2.2, Update Profile) and will then pass them to the UMS by calling its update function.

As mentioned earlier in chapter 6, the AGS replaced the pipeline in the former WHURLE. The AGS is responsible for managing, simplifying and coordinating the communication and flow of data as well as for reducing traffic within an adaptive service-oriented and distributed learning environment. The AGS's front-page, automatically created by NuSOAP, and is shown in the following figure (7.5):

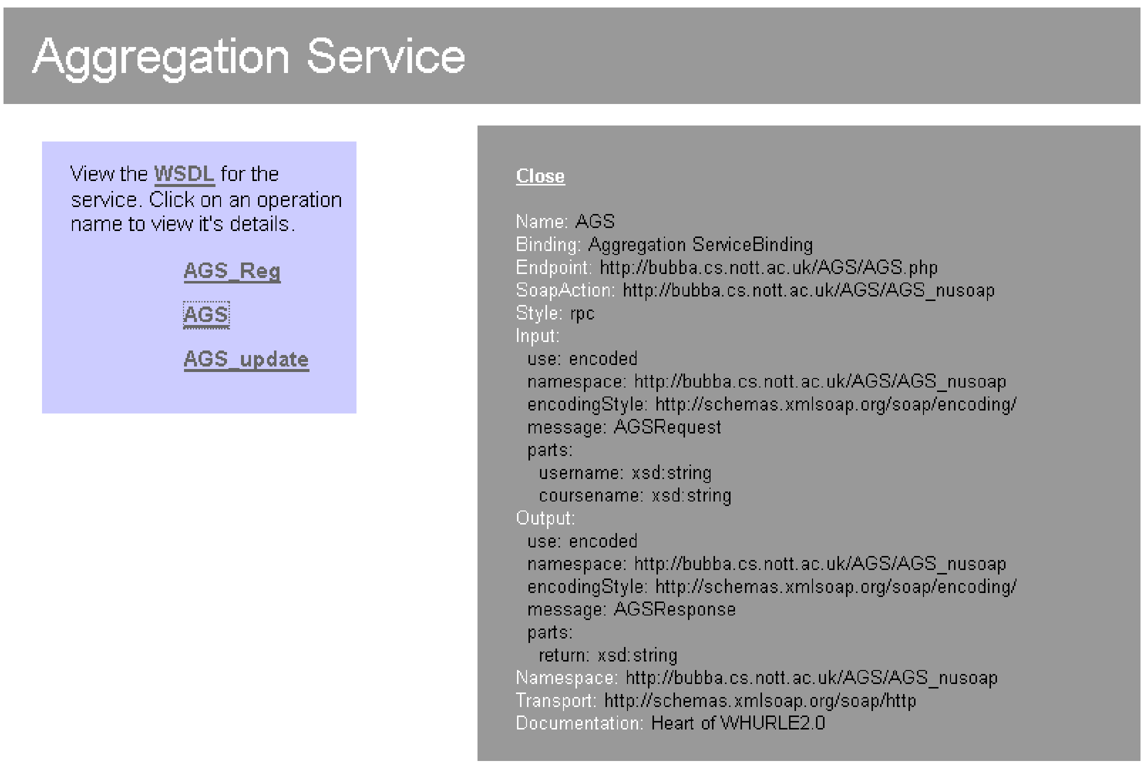

Figure 7.5: The Aggregation Service's Front-page, showing its three available functions and providing the description of one of those functions (right), which is the `AGS` function.

7.2.3 User Modelling Service (UMS)

As for the UMS, primarily for experimental purposes, a simple knowledge-based stereotype user-model is implemented, which has three possible story lines: Beginner, Intermediate and Advanced. However, the framework is designed to allow for future replacement by any other richer UMS. The service's has an XML database which stores the users' profiles, and is queried using XSLT:

1) to register students and assign to them levels according to their grades (referring to a declared scale) 2) to match the users with their levels of knowledge at runtime to decide upon the content with which they will be presented and 3) to update their level as the students progress within the course.

When registering a student to the UMS's database, the e-mail address is chosen as the unique identifier to avoid registering different users from different LMS's with the same username. This is due to the fact that a username is indeed unique in a single installation of Moodle but not in all its available versions, or in any other LMS such as ATutor (chapter 2, section 3.3.2). This could lead to ambiguous authentication issues, even if other parameters such as first and second names or password are also registered. The e-mail address acts as a unique identifier to refer to this particular learner. Hence the same content is provided to him/her in any other LMS that is integrated to WHURLE 2.0, as long as he/she registers with the same e-mail address.

Since only one course (or module) is made available through this implementation of the system, there was a need to identify only the user. However, if more than one course is provided by a system, then there will be a need to distinguish the user's level in a particular course, which might vary with different courses. There are at least two possible ways to achieve this: one is by providing the UMS with a database of user models for each course; another option is to modify the UMS database to include a course section with which the level is associated. Moreover, a student can have a different level for each lesson in the same course.

The UMS contains the following files:

- **UMS.xml**: this is the database that holds essential information about each student. It is queried by XSLT scripts to achieve one of the following: register a new student, determine the user's profile, or update this profile.
- **UMS.php**: implements the service by registering the three accessible functions of the UMS. In addition, it is responsible for handling all the communications with the AGS. The three functions of the UMS are:
  - **Register_User_AND_Assign_Level**: this function receives the `AGS_register` call (section 7.2.3) to register a new student and assign a new level for this new entry according to the attached grade, which is compared against a pre-defined scale (section 7.2.2).
  - **Get_User_Level**: this function is responsible for returning the level for a matched user, which is used for the selection of adaptive content. It receives a call from the `AGS` function as mentioned earlier.
  - **Update_User_level**: this function receives the AGS call through `AGS_update` and is responsible for comparing the new grade obtained by the learner, when he/she logs out of the system in order to update their current level in the UMS database. The user could sustain the same level, be upgraded to a higher level, or be downgraded to a lower level.
- **UMS1.xsl**: queries the database using the user's id which has been assigned by the function `Register_User_AND_Assign_Level`, if the id is matched, then the user already exists and does not need to be registered again. Otherwise, a new user profile is created in the UMS database.

- **UMS2.xsl**: this stylesheet is called by the function `Get_User_level` to query the UMS database to determine the user's level (for example Intermediate) which is then passed to UMS.php. From the UMS.php it is passed to the AGS to continue the adaptive content preparation process. An illustration showing the communication between the three scripts is given in figure (7.6).
- **UMS3.xsl**: this stylesheet is called by the function `Update_User_Level` and queries the database using the user's id; if the user exists then it will update his/her profile, otherwise it states that the user does not exist.

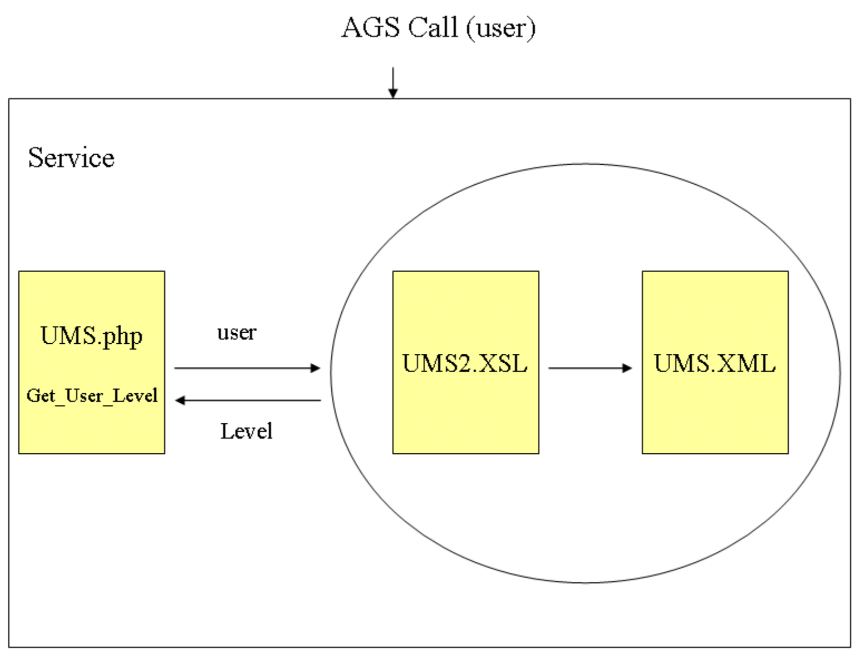

Figure 7.6: The User Modelling Service (UMS): the second function in AGS (*AGS*) calls the second function in UMS (*Get_User_Level*) in UMS.php, this function applies the UMS2.xsl to the UMS.xml database and returns the result to AGS.php.

7.2.4 Lesson Plan Service (LPS)

The development of the LPS is similar to that of the UMS. It has an XML database, where all the names and locations of the lesson plans are stored. The AGS passes the lesson's name to the LPS which will then return the lesson plan's URL to the AGS. The

LPS consists of the following files: LPS.php, LPSDB.xml, LPS.xsl and a number of lesson plan files (UML.xml for the Software Engineering Course). The first three files are almost typical of those mentioned in the previous section, the only difference being that the service here receives the lesson's name (instead of the username) and responds by returning the lesson plan (instead of the user's profile (level) in the UMS). In addition, the LPS works as a repository that stores the actual lesson plans for different courses which are used by the AFS for selecting the required chunks. A snapshot of a lesson plan can be seen in figure (7.7):

```
 <level name="BuildCD" title="Build Class Diagram">
        <page>
         <chunk level="Beginner">ClassDiagram3_Beg</chunk>
         <chunk level="Intermediate">ClassDiagram3_Beg</chunk>
         <chunk level="Advanced">ClassDiagram3_Beg</chunk>
         </page>
</level>
```

Figure 7.7: Lesson Plan Service (LPS): a lesson plan's snapshot.

As the AGS (`the AGS` function in particular) has those two significant arguments (the user's level and the lesson plan), it will pass them to the Adaptation Filter Service (AFS) presented in the next section.

7.2.5 Adaptation Filter Service (AFS)

The AFS applies a special XSLT file to the given lesson plan (an XML file) using the level parameter; both arguments (lesson plan and level) are passed to it through the AGS communications with the UMS and the LPS in order to apply the required adaptation. AFS filters the lesson plan according to its rules and produces a list of required chunks for this specific learner as specified by the teacher in the lesson plan. The AFS consists of two distinct files:

- **AFS.php**: implements the service and applies the adaptation rules. This script is responsible for accepting clients' calls and responding with the result when communicating with the AGS.
- **AFS.xsl**: the filter which implements the user's level (for example Intermediate) against the XML file (lesson plan) and returns the results of the transformation to the AFS.php.

When the AGS receives the AFS results, it stores them into another XML file that is created instantly. This XML list of required chunks is then passed to the Chunk Management Service (CMS) where the selection of those chunks takes place, as will be explained further in the next section. A snapshot of the WSDL[78] file (chapter 4, section 4.5) for the AFS, is shown in figure (7.8) below:

```
- <definitions targetNamespace="http://bubba.cs.nott.ac.uk/AFS/AFS_nusoap">
  - <types>
    - <xsd:schema targetNamespace="http://bubba.cs.nott.ac.uk/AFS/AFS_nusoap
        <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
        <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/"/>
      </xsd:schema>
    </types>
  - <message name="Get_Chunks_namesRequest">
      <part name="Level" type="xsd:string"/>
      <part name="LP" type="xsd:string"/>
    </message>
  - <message name="Get_Chunks_namesResponse">
      <part name="return" type="xsd:string"/>
    </message>
  - <portType name="AdaptionFilterServicePortType">
    - <operation name="Get_Chunks_names">
        <input message="tns:Get_Chunks_namesRequest"/>
        <output message="tns:Get_Chunks_namesResponse"/>
      </operation>
```

Figure 7.8: The Adaption Filter Service (AFS): a snapshot of the WSDL file

[78] WSDL: Web Service Description Language

7.2.6 Chunk Management Service (CMS)

Chunks (chapter 6, section 6.2.1), are XML files that contain text or references to other media types. This service consists of an XML database which contains important meta-data about the chunk such as its name, location, type, author, level of difficulty or any other meta-data information. This meta-data would help in the semantic exchange and interoperability of those chunks among different portals and LMS that choose to use them. Chunks classified as "internal" are stored in the CMS, which also serves as a repository for these.

The list of chunks produced by the AFS (see the previous section 7.2.6) is used to select those specified chunks and return them back to the AGS. The AGS then serves the chunks' content to Moodle's client (AGS_client2) as mentioned in section (7.2.2).

The CMS contains three main files in addition to the repositories of local (internal) chunks stored there. Those files are:

- **CMS.php**: this script implements the service and returns the results to the AGS at the end of the process.
- **CMSSDB.xml**: an XML database that contains all the required information about the chunks including their locations.
- **CMS.xsl**: applies specific transformation rules to the XML file received from the AGS to select the named chunks, extract their content and then return these results to CMS.php that passes this content back to the calling client (AGS).

- **Chunk files**: different XML files containing the actual learning content. Each chunk is a conceptually-discrete unit of information and one chunk or a group of chunks forms a single adaptive lesson.

For this specific implementation of WHURLE 2.0 presented throughout this thesis, it is assumed that all the chunks are stored locally. Chunks as well as the lesson plans for the Software Engineering module - presented in the next chapter - are developed manually by writing a number of different XML files, using simplified templates from the former WHURLE project (Brailsford 2002b). An example of a chunk is shown in figure (7.9):

```
<chunk type="text" name="ClassDiagram3_Beg.xml"  title=" Class Diagrams: Cont
       <versionlist>.
              <version date="2008-04-1">.
                     <author>.
                            <name>Maram Meccawy</name>.
                            <address>School of Computer Science &amp; IT,
                     </author>.
              </version>.
        </versionlist>.

       <text>.
              <p>Let us start to build some class diagrams</p>.
              <p>The class is represented by a rectangle</p>   .
              <p>It has a number of sections (usually 3).  The first contain
              <p>The general form of the class diagram is shown below</p>
       </text>.
       .
       <media type="JPG" title="Class Diagram" name="CD1" src="http://bubba.c
       </media>.

       .
       <keywords>.
                     <keyword>UML</keyword>.
                     <keyword>Software Engineering</keyword>.
                     <keyword>Class Diagram</keyword>.
       </keywords>.
</chunk>.
```

Figure 7.9: The Chunk Management Service: a snapshot of a Chunk

7.3 Summary

This chapter explained the physical implementation of the architectural design concepts that was described in the previous chapter. It provided a detailed documentation of the WHURLE 2.0 development including both Moodle's tools (in addition to the independent Web services. Moreover, Chapter 7 highlighted the arguments passed between different components explaining the mechanism of how they collaboratly functions in greater detail.

The test settings for evaluating the implemented learning environment are explained in the next chapter.

CHAPTER 8

"Knowledge must come through action; you can have no test which is not fanciful, save by trial"

Sophocles (486 BC - 406 BC)

An Evaluation Framework for WHURLE 2.0

8.1 Overview

Following the implementation of the WHURLE 2.0 learning environment (chapter 7), an evaluation framework was designed to test its relevance. This framework focused on the following aspects of the developed system: interoperability, reusability, flexibility and openness, robustness and stability, and collaborative learning features.

This chapter documents the evaluation process of the learning environment introduced in previous chapters. It then analyses the results obtained from those tests and evaluation methods. However, the main purpose of the evaluation framework was to test both interoperability and reusability in the context of WHURLE 2.0's learning environment as they are the two main issues addressed by this new architecture

8.2 Interoperability

8.2.1 Introduction

When the integration of adaptive learning services with the Learning Management System (LMS) Moodle (chapter 3, section 3.3.1) as a delivery system was decided, the next challenge was how to achieve this technically. There were a number of options available, such as: extending Moodle's own code, integrating the services into the LMS through a middleware, or using Web services' technologies. Many reasons have influenced the choice of the latter as means of communications, among them the fact that Web services allow easy integration since they are designed to deal with interoperability issues across manufacturers, platforms, programming languages and so forth. WHURLE 2.0 is a framework, not a system, and therefore testing its granularity and applicability regardless of which LMS (delivery and interface portal) is used, is an important part of its evaluation.

The WHURLE 2.0 architecture equation can be described as: Adaptive Educational Hypermedia (AEH) capabilities deployed as a set of collaborative Web services accessed via a general purpose Learning Management System. So far, Moodle has represented the LMS in this equation, while in theory any LMS is capable of replacing it. Therefore, ATutor (chapter 3, section 3.3.2) was chosen for this specific experiment to replace Moodle as the delivery system.

8.2.2 Methodology

The aim was to test the system's integration of the adaptive services with ATutor. The question here was: is WHURLE 2.0 framework generic and interoperable as it has claimed, and to what extent?

Before this experiment took place, a case study for user trials (described later in this chapter) was defined. This case study involved teaching a Software Engineering module. For the experiment with ATutor to be more accurate, the module structure and supporting tools in Moodle were replicated to represent an identical module in ATutor.

Since the students were already registered with the User Modelling Service (UMS) - this process is described in detail in the case study in section 8.5 - and no students were using the system in this experiment, the first and third Web service clients (chapter 7) were not needed. The focus was on modifying Client2 that communicates with the AGS by providing the unique identifier (the user's e-mail address) as well as the current lesson's name to receive the adaptive and personalised lessons. A new client was written and installed in ATutor; this new version of Client2 resembled to a great extent the client needed for Moodle, the only differences being in the names of variables that are LMS-specific[79].

Since Moodle and ATutor have similar database structures and authentication methods, the main challenge was to identify the names of variables and their location for the client's parameters within each LMS. For example, in Moodle the username is stored as *username* in table *mdl_user* while it is stored as *login* in *AT_user* in ATutor.

Another challenge was to decide where this adaptive content would be presented i.e. which tool or feature should be used? In ATutor the course or modules has no *Lesson activity* but it has what it calls a *content page*, which is the main page for the courses with its own navigational path. Hence each link or page of this content could replace the Lesson activity in Moodle.

The next step was to identify where to store the adaptive lessons prepared by the Web services. These would be stored in ATutor's database. In Moodle they were stored in table *mdl_lesson_pages* under the *content* field while in ATutor they were stored in the *AT_content* table under the *text* field.

[79] See Appendix for a version of each client

The following screenshots show first how the links of the adaptive lessons (content navigation) appeared on ATutor's main page in figure (8.1) and then an adapted lesson delivered in ATutor is presented in figure (8.2):

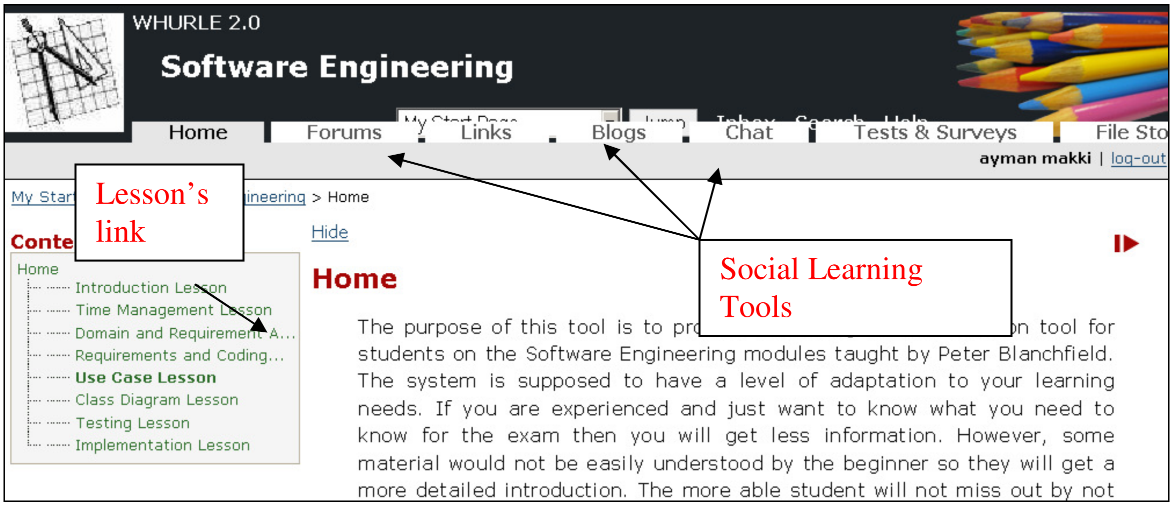

Figure 8.1: WHURLE 2.0 Learning Environment: ATutor as the delivery service that provides both adaptive learning content (through *Content* feature indicated by the small arrow) and social tools.

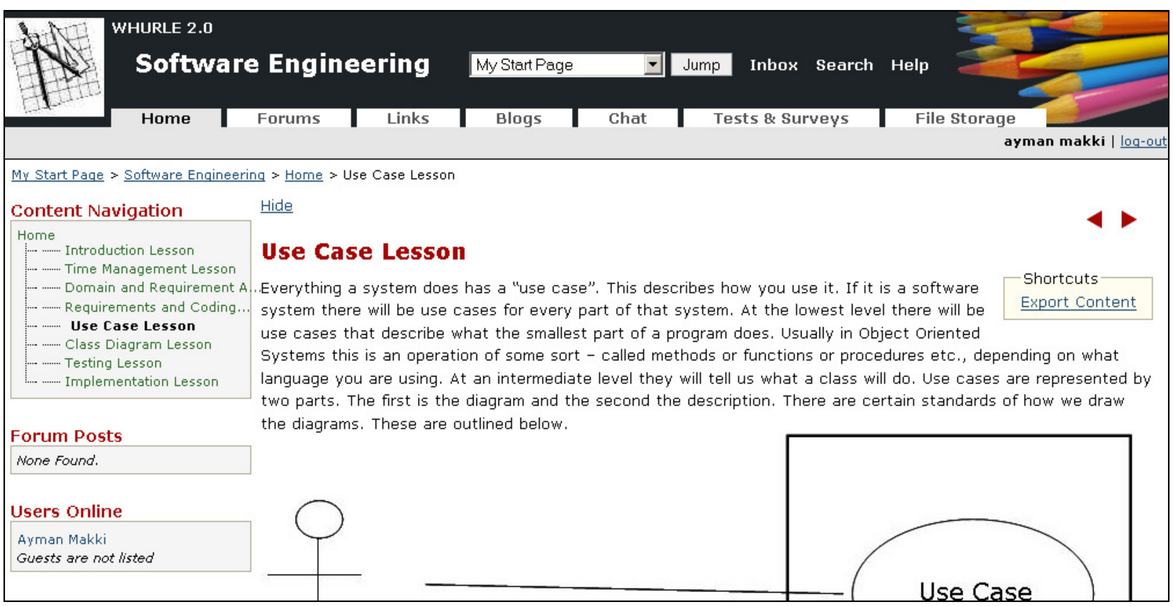

Figure 8.2: An adaptive Lesson about *Use Cases* as it appears for a learner that has been modelled as a **Beginner** Learner in ATutor (compare it with the same content presented in Moodle in figure 8.5.b).

8.2.3 Results

This experiment showed that the integration of the adaptation services with another LMS was not only achievable, but also smooth requiring very little effort and time. The Web services which performed the adaptation needed to be developed only once; when the second LMS (ATutor) needed to access the learning content, the only part that required a slight modification was the LMS's client. This high level of interoperability was achieved by the use of Web services. Therefore, the integration of services into ATutor showed that the framework was largely generic and thus WHURLE 2.0 architecture catered for interoperability of the adaptation functionality. The user was no longer locked in one adaptive system; in fact, a student could be taking two adaptive courses in two different LMS at the same time, using the same adaptation services.

The results of the experiment also confirmed the suitability of open source LMS to be used as delivery systems for adaptive learning content. This is due to their technical flexibility which allows them to be extended or integrated with external independently-developed software, good documentation and the free support of the LMS online community. In addition, the experiment showed the importance of having a broad model for the Learning Management Systems. The resemblances between the database structures, authentication method and social tools in both Moodle and ATutor saved time and effort when the integration process was repeated. However the most difficult part with the second integration was to find the variable names. Moodle has existed longer than its competitor and therefore has better available documentation than ATutor, while installing ATutor was less challenging than setting up Moodle.

The experiment has also revealed an important point regarding presentation and how each individual LMS was designed. Presentation has an impact on the end users and hence however effective the adaptation, if it is poorly presented it will dissatisfy the users. It is important, therefore, to choose the right tool in an LMS to deliver the adaptation, provided that other choices are available, as was the case in Moodle.

Finally, as mentioned earlier, evaluating the interoperability in WHURLE 2.0 was the key goal in this experiment and this experiment showed positive results regarding this issue.

8.3 Reusability

Testing the reusability in the context of WHURLE 2.0 was also achieved through the experiment explained in the previous section (8.2) and in the case study that will be described in section (8.5). This section (8.3) emphasises the importance of the issues involved.

Reusability in the context of AEHS can be investigated from different angles: reusability of content; reusability of the components of the system; and reusability user profiles, as will be explained shortly.

The content described in the case study (section 8.5), was developed for that specific module that has been taught in the University of Nottingham. However, the decision to service the content by dedicated Content Management Service (CMS) meant that it could be launched by any LMS and therefore be fully or partially reused. Since the content was not hardwired into the system, it could be reused independently. Nevertheless, the semantics of the content would need to be reasoned if the same content was to be restructured or reused as part of a different module. This showed the importance of the CMS databases that provided the metadata for this content. The experiment with ATutor demonstrated that the same content could be easily immigrated from one learning environment to another and any updates made to such content were immediately reflected in all the systems that used it. This is due to the content being served instantly during run-time in WHURLE 2.0 and not copied and pasted or uploaded to each delivery system.

The same was true for the user profile which was available through a dedicated service. The user model and the students' profiles were reused by another delivery system, ATutor. Even when the content was changed, ATutor was still able to communicate with User Modelling Service (UMS) via Aggregation Service (AGS). In fact, since UMS is a service in its own right, a client that sought only to retrieve users' profiles was able to achieve its goal by communicating with UMS directly.

Since both CMS and UMS were also components of the adaptive system, the ability to use them independently and the ability to communicate with them either directly via a Web service client or through the AGS also demonstrated the ability to reuse such components at a technical level. More discussion regarding this issue at semantic levels is presented in chapter 9 (section 9.1).

8.4 System's Flexibility and Openness

The framework was also interested in testing the extent of architecture flexibility and openness. The questions were: how flexible and open is this architecture? How difficult is it to add a new service, for example?

Those questions were answered through the integration of a new authoring aid known as WHURLE2.0Author (Obeidat 2008), which was developed independently. This authoring aid was designed for creating chunks and lesson plans automatically. It was integrated into the system's architecture, as can be seen in figure (8.3) below.

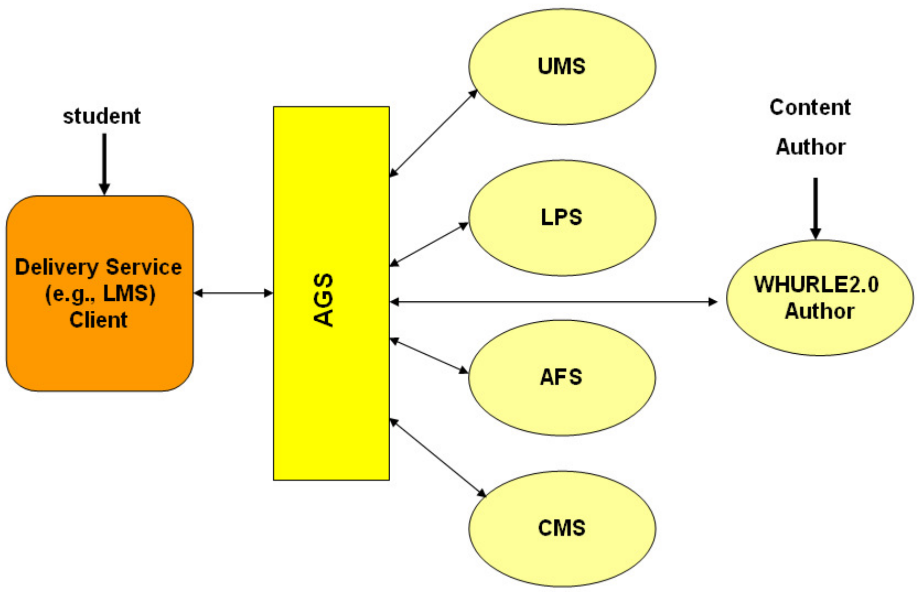

Figure 8.3: An Authoring service added to the WHURLE 2.0 architecture.

It can be noticed from figure (8.3) that the design of this new authoring service did not interfere with any of WHURLE 2.0's components. Rather, it co-existed with the adaptation services by communicating with the Aggregation Service (AGS), which dealt with different requests and responses from clients-services, and managed WHURLE 2.0 internal and external communications. As explained earlier in chapter 6, the AGS – introduced in this service-oriented version of the WHURLE framework- acts uniquely both as a client and a service.

The authoring service for WHURLE 2.0 communicated with the AGS either to provide authored-edited chunk and lesson plans that would be then passed to their dedicated services, or to accept chunks and lesson plans from AGS to be edited when required.

The integration of the new service into WHURLE 2.0's architecture was highly feasible as a result of the decision to have a dedicated service for integration and traffic control and to use the technologies and standards of Web services which set interoperability between independently-developed software as its main goal. Hence this opened the architecture to collaborate with external services/components and allowed for greater flexibility of the WHURLE 2.0 Framework.

8.5 Robustness and Stability

8.5.1 Introduction to the Case Study

A few users testing the system could have demonstrated whether or not it functioned. However, a trial was conducted in which the system was used in a real world situation with potential users. It aimed to assure the validity of the architecture at an enterprise level; would the system operate and deliver its intended learning content to the designated students when they all log into it synchronously?

The WHURLE 2.0 learning environment was used by a total of 140 students from different groups at the University of Nottingham: first-year Computer Science students following an introductory Software Engineering (SE) module; and Masters' students ,following a similar software engineering module as part of a number of Computer Science and IT courses. The latter group included students from different first degree backgrounds, diverse nationalities and languages, and varying levels of knowledge in

Computer Science, representing the ‘real-world’ in terms of the need to apply adaptation and personalised learning.

8.5.2 Methodology

In Moodle, the module was organised into eleven sections with defined topics; the first eight topics or sections introduced a specific concept of Software Engineering, such as Requirement Analysis, Design, Uses Cases and Testing. The final section enabled a Post-test and a user-satisfaction questionnaire.

The system was given to the students as a revision guide five weeks before the exam; they were asked to register with the system in their own time. It was an optional study-aiding tool since students had access to other resources including lectures, notes, slides and books on the course’s reading list.

The students first registered individually with the Moodle LMS through its built-in registration form. When a student accessed Moodle, he/she was given a Pre-Test using Moodle’s *Quiz* activity tool (chapters 3, section 3.3.1), which could be taken only once during this experiment. The Pre-Test is composed of seven single-choice questions. The answers were measured against a scale of 1-9 marks, where scoring 1-3 indicated a *Beginner*, 4-6 an *Intermediate* and 7-9 an *Advanced* type of learner. The Pre-Test was prepared by a field expert who taught the module. Assigning students’ levels according to their test results was considered a more accurate method of assessing their knowledge, rather than assigning levels manually using the tutors’ prior experience of their students, or referring to the students’ previous results. However, different teaching scenarios require different methods for assessing students’ levels.

After the students had received feedback for their answers to each question, they returned to the course's main page and clicked on a *Resource* link (chapter 3, section 3.3.1), which triggered the client that called the AGS. This client-call registered the students with the external User Modelling Service (chapters 7) and assigned a level of knowledge according to their Pre-Test results. Students immediately received feedback stating their score for the test and the corresponding level that they have just been assigned as can be seen in figure (8.4).

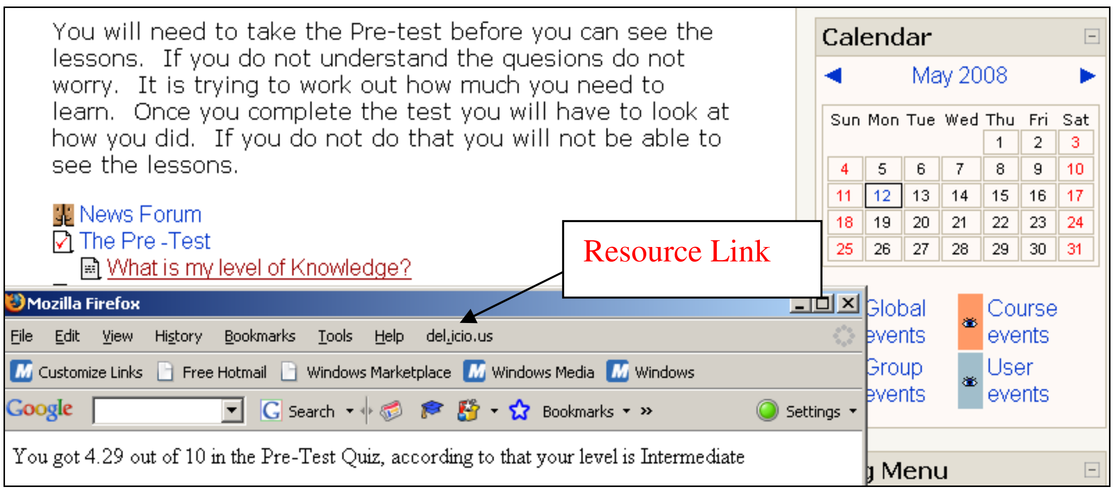

Figure 8.4: Registering to the backend User Modelling Service (UMS) by clicking on the Moodle's Resource link

The *Lesson* activity (chapter 3, section 3.3.1) was used to provide the personalised learning content in Moodle. In the last section of Moodle's course, students were given a Post-Test again using the Quiz activity tool, this was done to firstly, motivate students to use the system during the revision period. Secondly, this quiz was also used for profile updates. However, because of the relatively short period of this experiment, the changes in user profile were not significant and so no great attention was given to such updates in this particular case study. Below are two screenshots from WHURLE 2.0 showing the same lesson adapted for both Advanced and Beginner level students:

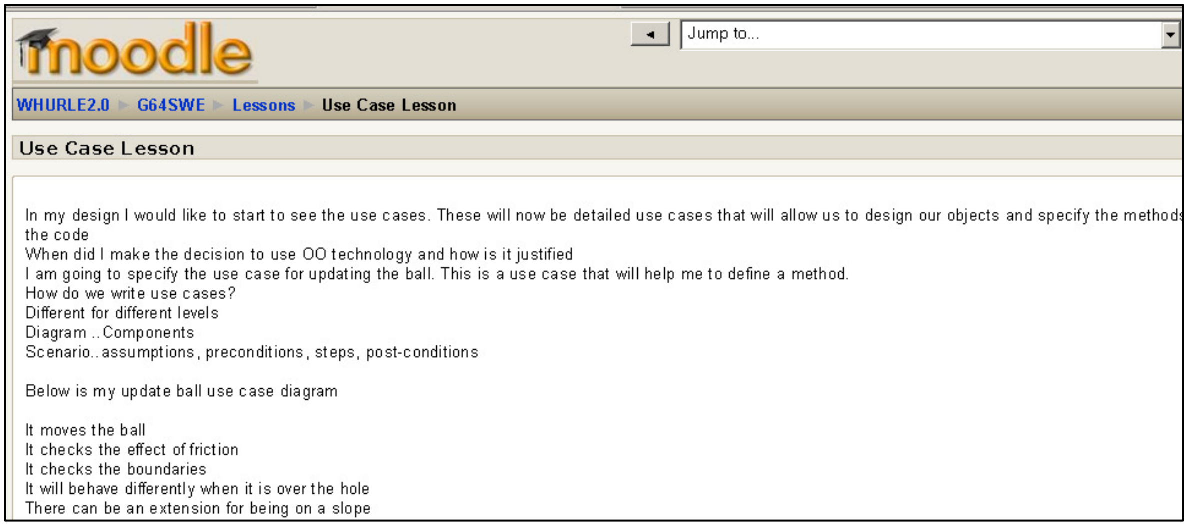

Figure 8.5 a: An adaptive Lesson about *Use Cases* as it appears for a learner that has been modelled (after taking the Pre-Test) as **Advanced**.

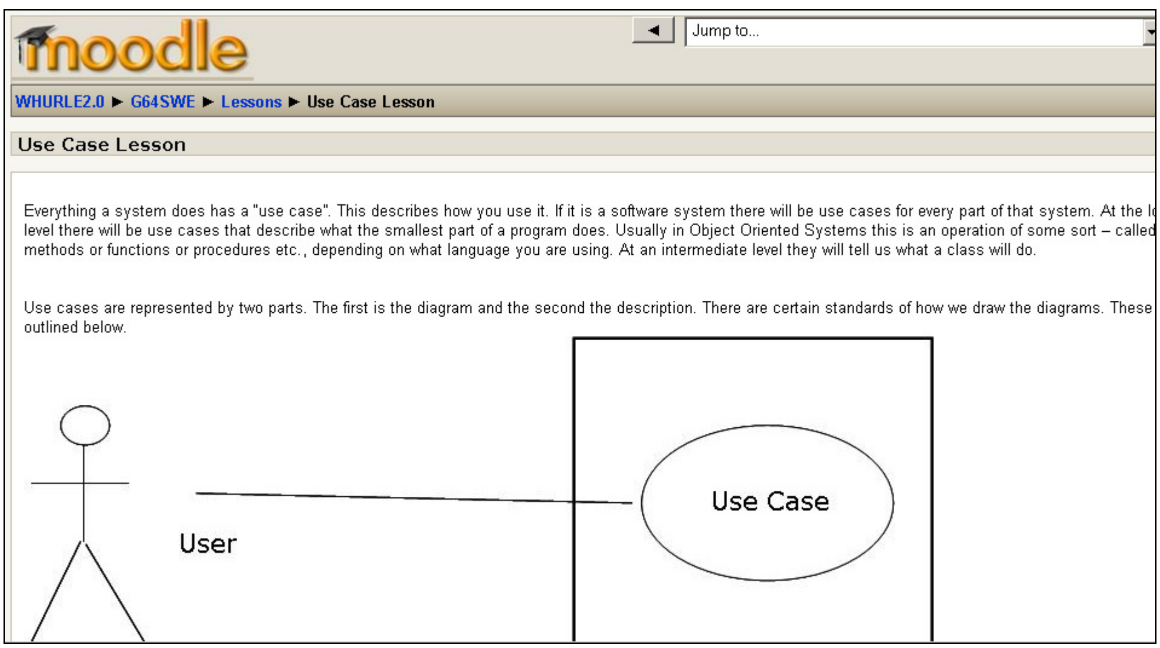

Figure 8.5 b: An adaptive Lesson about *Use Cases* as it appears for a learner that has been modelled as **a Beginner** Learner and so would have more introductory information and content than the Advanced learner in the preceding figure.

Since the personalised content is prepared in real time, the Lesson activity could fail to show any learning content for various reasons, such as the students not registering correctly with the User Modelling Service (UMS). A risk-control procedure was therefore taken, using Moodle's *Book* activity (chapter 3, section 3.3.1). In this Book, a default (Beginner) view of the learning content was provide; this could be used by the students as a static available reference. In addition, it served another purpose, since the Advanced user could obtain different information from the Beginner (often less as in this implementation); having a complete version available to him/her could also address possible complaints about not having the full picture, or about missing vital information. Moreover, it allowed an Advanced user to compare the two versions to decide which was more suitable for his/her learning needs, thus providing useful feedback.

8.5.3 Results

The goal of this case study was to find answers to the following question: did the system perform as intended and expected in the real world settings?

The answer was that the system not only functioned effectively but also handled this large number of students efficiently; the adaptation was achieved and the components of the framework worked collaboratively and smoothly with one another. Moreover, there was no delay in presenting results compared with the previous monolithic WHURLE, taking into consideration that the type of content presented to the students was limited to text and figures. The students did not report any noticeable delay in starting their lessons. The integration was transparent to the users since the learning content was presented to them using Moodle's standard tools.

Moodle, along with most LMS, was easy to use and did not require any training for either tutor or students, which is not the case for a traditional Adaptive Educational Hypermedia System (AEHS), thus justifying the use of LMS as a delivery system. In addition, the LMS provided the users, especially teachers and administrator, with an integrated learning environment in which many teaching and administrative functions were available through a single interface. Adaptation from the services in addition to the functionality provided by LMS were all integrated in one system. This addressed the lack of integration issues in the majority of AEHS as well as enriching LMS with the adaptation functionality.

8.6 Collaborative e-Learning Enabled via WHURLE 2.0

8.6.1 Introduction

The goal of the case study (introduced in section 8.5) was to find answers for two questions; the first one was answered in the previous section. The other question focused on the social and collaborative aspects of WHURLE 2.0: did the students actually use any of the enabled social and collaborative learning tools?

This question was asked since one of the criticisms faced by AEHS is that they are single- purpose systems that lack social learning functionalities compared to LMS. One of the reasons, therefore, for choosing an LMS as delivery system/service was to present a complete modern online learning environment that is capable of integrating personalised learning with e-Learning 2.0 tools and technologies.

8.6.2 Methodology

As mentioned earlier in the description of the case study (section 8.5.2), the Software Engineering (SE) module had been organised into eleven sections with defined topics that were presented as adaptive lessons. In addition, each topic used a number of the built-in tools of the LMS Moodle that aimed to enrich and improve the learning process where appropriate, these tools included: Forum, Lesson, Resource, Chat, Quiz, Wiki, and Book. A snapshot of the environment with the different sections and tools is shown in figure (8.7).

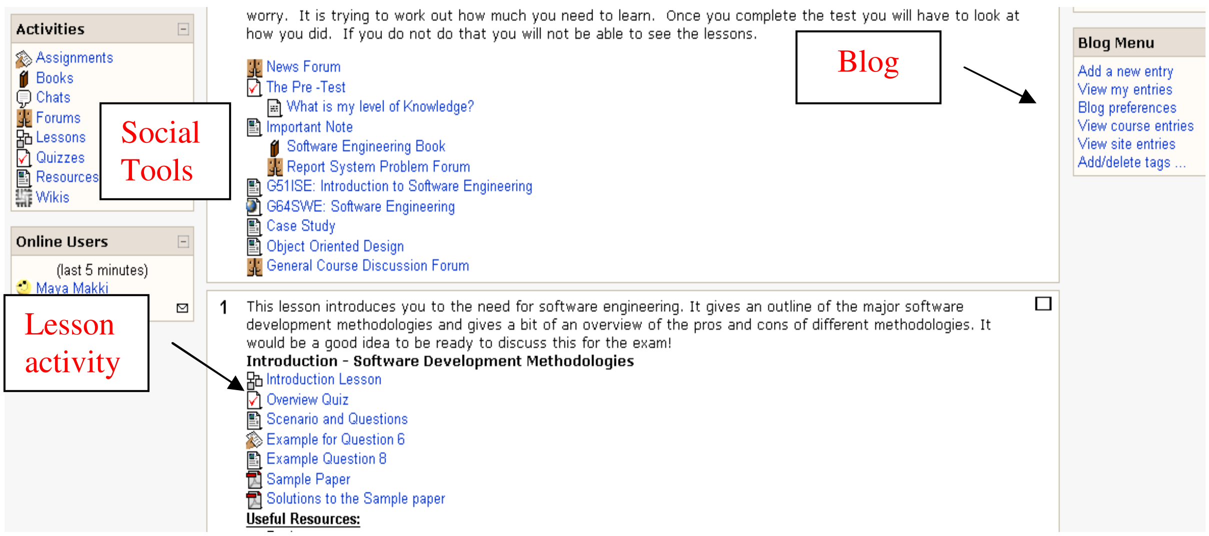

Figure 8.6: WHURLE 2.0 Learning Environment: Moodle as the delivery system that provides both adaptive learning content and social tools.

The results described in the following section were obtained from Moodle's activity report and an online questionnaire enabled through Moodle's links to external resources

8.6.3 Results

Moodle activity reports showed that the system had been heavily used in the five-week period. It was also reported that some of the students in this case study often use a number of social networks and collaborative or communication tools in their daily life and for educational purposes; they agreed such tools were important and useful. However, they did not use them to a large extent in this case study. The reasons given included: limited period, exam preparation and not enough participants at any one time to communicate with in the case of some tools such as the Chat.

Other factors that were observed during this case study included the student's level of maturity and cultural background. Master students freely entered the online discussion and as a result evidence of this discussion showed up in their exam responses. On the other hand, the undergraduates did not take part in the discussion and the responses showed that the students had not considered these areas before trying to answer the exam questions. Therefore, it is felt that it would be a good opportunity for more mature groups to be offered the chance for more social learning, while the less experienced group is given more directed teaching. However, maturity is not the only factor leading to differences in collaboration between the two groups. The students' background could also have affected this. For example the majority of undergraduate students were home-students, while the majority of the postgraduate students were international students, and so the level of collaboration could vary from one culture to another. More investigation using qualitative research methods such as interviews might therefore be necessary to understand this issue of low level of online collaboration and to draw conclusions supported by concrete evidence.

Nevertheless, the fact that not many students used those tools in this case study should not under-estimate their benefits or undermine its potential capability for e-learning. It would be useful to observe the usage of those tools over a longer period - for example, a whole semester.

The important conclusion from this case study regarding the social or collaborative learning side of WHURLE 2.0 is that the adaptation services did not inhibit the use of the

LMS's activities; in fact, they enhanced the LMS's capability without negatively affecting the LMS main features.

The case study has also helped to provide some suggestions about how to address this poor usage of collaborative tools using adaptation techniques, this might result in having group-adaptation, as can be seen in the next section.

8.6.4 Collaborative Learning and Group-Adaptation: A Brief Discussion

In terms of the collaborative and social learning, the system tried to resemble the classroom environment where students were actively socialising and participating with one another in the learning process. However, the results of the case study did not meet expectations. They did not create the rich environment that would have allowed them to benefit from one another's knowledge and experiences.

Despite the reasons given earlier in section (8.6.3) by the students to justify this low level of participation in these online social activities, further investigation is needed. In addition, there is another question: how can the knowledge from individual adaptation and user modelling help in applying adaptation techniques to bridge this gap and enhance student collaboration?

In Moodle, a lot of information in the database tables has been collected from the users' interaction with the different solo or social activities presented to them within this environment. Locating this information and transferring it to the UMS does not present a real challenge compared with inferring underlying meanings both for individual and group adaptation. The meaning of the data needs to be investigated in greater detail. The same concept could be applied at a group level *i.e.* can it be assumed that students who participate in the same activity are at the same knowledge-level? What if one student was actually asking a question while the other was providing possible answers? This might indicate that the second one had more knowledge or had spent more time revising. Does it reflect a sharing of the same interest? Again, this may not necessarily be the case, since it could be that the student was posting a negative comment regarding the topic being discussed.

Nevertheless, a first step could be taken by monitoring students who participate in, for example, more than one activity, or more than once in the same activity, and who could then initially be placed in the same group. Hence, their user profiles could be compared, and when one student engages in a new activity, this activity is recommended to the others.

Finally, as mentioned earlier in chapter 3, group-adaptation should not replace individual adaptation, since students remain unique and group adaptation could be regarded as another dimension for addressing students' learning needs. The research presented in this thesis does not yet adapt to groups, however, it takes the first step towards this goal.

8.7 Summary

The research presented in this thesis was intended neither to argue the benefits of adaptation nor to favour a certain adaptation method or technique. It aimed to remove the barriers and limitations that are holding back Adaptive Educational Hypermedia Systems in order to promote their use in the mainstream of Web-based learning. Only then can these systems be examined further to determine their true benefits.

This chapter has described the evaluation framework that was defined to test the relevance of the proposed architecture known as WHURLE 2.0. The results obtained from the evaluation can be summarised as follows:

- WHURLE 2.0 architecture achieved high levels of: interoperability, reusability, flexibility and openness.
- The integrated adaptive and collaborative learning environment functioned efficiently and effectively in real world settings.
- The collaborative tools of this newly-integrated learning environment enhanced the adaptive learning experiences without negatively affecting Learning Management System and *vice versa*.
- The case study showed that students did not make use of the social and collaborative learning tools to the level that was anticipated.

The next chapter elaborates on those findings and results, summarises the whole thesis and draws its final conclusions and recommendations for future work.

CHAPTER 9

"Success is a journey, not a destination. The doing is often more important than the outcome"

Arthur Ashe (1943-1993)

DISCUSSION & CONCLUSION

9.1 Discussion

The research presented in this thesis aimed at dealing with the limitations surrounding adaptive systems in the context of the requirements of modern e-learning and hence at promoting their use outside research laboratories. The analysis of the literature has suggested that the shortcomings identified should be addressed at an architectural level. The research presented thought this thesis therefore has looked into re-engineering the architectures of Adaptive Educational Hypermedia Systems (AEHS). This has resulted in the design of an online adaptive learning environment that is capable of overcoming the limitations of the existing systems.

The new architecture known as WHURLE 2.0 reused of the adaptation concepts of the former adaptive system known as WHURLE (Brailsford 2002b) and addressed some of its limitations. The monolithic system was turned into an open and distributed service-oriented architecture (SOA) in which its adaptation components became loosely-coupled services that communicated with one another through a coordinator service. The new WHURLE 2.0 architecture is implemented as a set of Web services using XML, SOAP and WSDL. Moreover, the new architecture allows the integration of any open source

Learning Management System (LMS) with all its desired tools and features, for example: Moodle (Moodle 2008), SAKAI (SAKAI)or ATutor (ATutor 2008), to be used as the delivery service.

One of the important limitations of AEHS to which the research has paid great attention is the issue of interoperability. Solving the problem of interoperability would also lead to allowing the reusability of content, user's data and the adaptive system's components. As mentioned in chapter 2 (section 2.5), this thesis classifies research dealing with the problem of interoperability and reusability in AEHS into four different categories: the Conversion approach, the *Lingua-franca* approach, the Universal Language approach and the Language-Specifier approach.

The architectural solution described here uses Web services languages and therefore follows the latter Language-Specifier approach. This decision has been made after examining other approaches and their respective limitations. The direct translation in the Conversion approach is more accurate in general but it is also more laborious to implement, with a translation required for every pair of interoperating systems. An example of this approach is the MOT-WHURLE translation (Stewart *et al.* 2004). However, this approach can be enhanced further using Web services technologies as shown by the case study presented in chapter 5 (section 5.2). As for the *Lingua Franca* approach, this may be the simplest solution, but is not necessarily the best since there is potential for attenuation of meaning over repeated or transitive translations; an example of this approach is seen in CAF (Cristea 2005). On the other hand, a Universal Language which does not yet exist for AEHS, could be seen as including both Semantic Web proposals such as Personalized Services (Dolog *et al.* 2004) (using ontology) as well as standardised solutions such as APeLS (Conlan *et al.* 2002). One of the problems with ontology, however, is that it is difficult to get different research groups to agree on the same definitions. In addition, it can be limited in capability, especially when too many features or unsupported features are added to the language, which make it unreasonably large. The same is true for standards: the currently-available e-learning standards are limited when it comes to adaptive, interactive or rich content (Rey López *et al.* 2007). Hence expansions and extensions are usually required to suit a different need, which in turn defeats, to some extent, the whole concept of an agreed standard.

On the other hand, a Language-Specifier, such as XML, is a more flexible and lightweight approach. It prescribes how each application's language should be interpreted and accessed at least on a technical level, although not necessarily on a semantic or interdependency level. The Web services' approach, with its XML base and languages is seen as the ideal way to address this problem within a SOA, where each component becomes a Web service that can be accessed and maintained independently. Decomposing the adaptation functionalities of an adaptive system into independent Web services, as in WHURLE 2.0, is therefore considered to be more efficient than wrapping the whole system as one Web service. Other systems, which have opened their systems but used original protocol instead of Web services to describe how the system should be accessed, can be considered as following this Language-Specifier approach such as in ADAPT^2 (Brusilovsky 2005). However, using a system-specific protocol does not offer a practical solution for real-world settings since many human communications are required throughout the integration process between content developer/provider and the delivery system's provider as demonstrated in the example given in case study 2, chapter 5 (section 5.3), while Web services automate this process from a technical prospective.

In WHURLE 2.0, in order to deal with the interoperability problem semantically, some design decisions were made - for example, the Chunk Management Service (CMS) is introduced (chapter 6, section 6.3.6). The CMS's XML database contains meta-data information about the chunks, hence moving to a higher level of interoperability. While the languages and descriptions of Web services allow easy access to the learning content (chunks) technically, this meta-data description tries to deal with the semantic aspect of the problem.

Moreover, the User Modelling Service (UMS) is designed following a centralised approach that has the advantage of a distributed user modelling service architecture which provides portable XML user profiles. End-users can therefore access learning applications offered through different LMS by different vendors reusing the information stored in their profiles.

In addition, the architecture focuses on separating the learning content and the user information from the adaptation rules or strategy, conceptually and physically. This allows any independently-developed portals (or delivery services) to access the learning content and the user profiles more easily.

Hence the aforementioned features of WHURLE 2.0 provides reusability of content, user profiles as well as the adaptation components given that the CMS and the UMS are also two of the components that comprises the new architecture.

As mentioned in earlier chapters, interoperability is needed on three levels: technical, which is achieved through Web services and XML; structural, achieved by re-designing the architecture and decomposing its components into independent services; and semantic, which is dealt with partially in CMS and UMS with their XML databases. However, they still require further investigation since the issue of interoperability for Adaptive Educational Hypermedia requires more than just dealing with interoperability of the learning content. Adaptive hypermedia systems comprise five layers as described in the LAOS model (Cristea and De Mooij 2003), which captures the interdependency of the five layers: content, user, adaptation, goals and constraints and presentation. It is not sufficient, therefore, to interoperate the content without simultaneously doing the same to the adaptation rules. In turn, the adaptation rules rely upon other aspects of the adaptive system. A change to a user model variable can affect outcomes of adaptation rules, which in turn modify the content and/or its presentation; at the same time, changes to user model variables can modify the goals and constraints. This interdependency of components distinguishes the problem of interoperability of AEHS from the much simpler problem of content interoperability. Migrating between AEH systems requires an adequate migration of all interdependent data, not just the content or user model. A Web services approach can support the migration of interdependent facilities.

Choosing an LMS to provide a delivery service and a login point for the integrated Web-based learning environment presents a two-fold solution. First, it enriches an Adaptive Educational Hypermedia System with social tools needed for modern learning experiences. Second, this integration results in utilising the LMS with an external transparent adaptation layer. It can be argued that instead of trying to convince institutes and teachers to switch to a new system in order to benefit from the adaptation, the latter is simply brought into their existing systems.

A number of researches (chapter 3, section 3.4) have realised the benefits of integrating LMS and AEHS, hence they have proposed different approaches to achieve this goal. However, they have extended a typical LMS and modified the system's codebase (Graf 2005). This process not only requires a thorough knowledge of the LMS's code but it limits the adaptation for this particular system. The whole process needs to be replicated for every new LMS that seeks to make use of the adaptation, resulting in a situation which resembles re-inventing the wheel. On the other hand, with the external Web service layer approach in WHURLE 2.0, two different LMS were able to access and receive that same adaptive content as seen by the experiments of using Moodle and ATutor in chapter 8. As a result, high levels of interoperability and flexibility is achieved by using Web services.

The results of the WHURLE 2.0 evaluation framework have revealed that the implemented system functioned as intended and the services collaborated smoothly. In addition, the flexibility and openness nature of WHURLE 2.0 allows any new service to be added to the architecture without affecting the other services. For example, a new authoring aid for creating chunks and lesson plans (Obeidat 2008) was developed independently and was easily plugged into the system. This is feasible because of the decision to have a dedicated service for integration and traffic control, the Aggregation Service (AGS), that acts both as a client and a service and is unique for WHURLE 2.0.

Moreover, the collaborative and social aspects of the system were examined, revealing a poor usage of the collaborating and Web 2.0 tools. The case study has suggested how to address this issue using adaptation, and this could gradually result in providing group-level adaptation in addition to individual-level adaptation.

9.2 Summarised Contribution

This thesis presents an innovative Web service-oriented architecture for an adaptive e-learning framework known as WHURLE 2.0. The architecture's flexibility allows for the adaptive learning content to be delivered in a Learning Management System, which has a number of desired tools, activities and functionalities to accommodate the needs of different stakeholders of the e-learning process. Moreover, the evaluation framework of this new architecture has shown positive results in terms of interoperability, reusability, openness and robustness.

A new classification is also introduced by this thesis which groups various proposals for solving the interoperability problem in the context of adaptive Web-based learning. The four categories are: the Conversion approach, the *Lingua-franca* approach, the Universal language approach and the Languages-specifier approach.

Two different methods for interoperability and reusability of learning content in adaptive systems have been investigated. The first method has suggested that Web services, when applied to the conversion approach, resulted in more flexibility and accessibility. The second method explored the reusability of rich adaptive or interactive content. This method has shown the complexity of interoperability of this type of content. In addition, it has highlighted that that the use of proprietary (original) protocols might not be a feasible solution for interoperability.

Finally, the evaluation case study has shown a low degree of interaction among students using the available Web 2.0 or social learning tools.

To date, 8 peer-reviewed publications have arisen from this research (See page XII).

9.3 Remaining Issues and Future Work

The case study evaluation was limited due to time constraints. Although a reasonable number of students participated in the case study and used the system over a period of four to five weeks, ideally the learning environment would have been given to the students at the beginning of the semester, with the adaptive lessons explained gradually during lectures. Better monitoring of the use of lessons and the social aspects of the system could have been better achieved if the system had been used over three months; more general conclusions could have been made. However, the LMS, namely Moodle as well as the online questionnaire, provided valuable information regarding the systems' usage and students' feedback that justifies different results or choices. This data could be analysed further by applying some qualitative research methods, including interviewing those students and reaching a better understanding of their responses.

This implementation of WHURLE 2.0 does not yet adapt to social aspects of the LMS nor does it adapt to different groups. However, the research presented here takes the first step towards achieving such goals: it brings adaptation to the environment where those activities take place, hence laying the foundation for group adaptation.

Creating chunks and lesson plans manually as XML files was a tedious and time-consuming process. Moreover, it is a complex process for somebody with limited IT knowledge of XML. An authoring tool should be developed to make this process easier. This tool could be made into a service that can be easily integrated into the architecture and hence have the resulting files transferred directly to the Lesson Plan service (LPS) and the Chunk Management Service (CMS). The authoring aid described earlier (Obeidat 2008) has laid the foundation for this tool.

Although the vast majority of LMS have no user modelling capabilities and lack a single dedicated location where all the information about a specific user is found, the system database itself contains a lot of distributed data about users. This information has been collected from the user's interaction with the different solo or social activities presented to him/her within this environment. One of the future plans, therefore, is to use some of this available information for user modelling purposes and to introduce a complete user profile or model to the LMS. However, the underlying significance of some of the data needs to be investigated in greater detail. For example, what does it mean (in terms of user's knowledge-level) if a specific learner is engaged in more than one chat room or if he/she is participating in a workshop or discussion? Could it be an indication of interest and progression? Or is it rather an indication that the student is facing a lot of problems and difficulties in understanding the course's or module's material? These are important questions that are yet to be answered.

9.4 Conclusion

The motivation to conduct the research presented in this thesis has come from realising the under-representation of Adaptive Educational Hypermedia Systems in the world of Web-based learning. A number of limitations and shortcomings have been proposed by the the AEH community as reasons for this absence, among them the limited reusability and interoperability provided by the adaptive systems. The work presented in this thesis attempted to address these limitations at an architectural level. This has been achieved by re-engineering an Adaptive Educational Hypermedia System (namely the WHURLE framework) by decomposing its architectural components into independent service using

Web services technologies. The result is the creation of an architecture which answers the needs of modern e-learning; this architecture was named WHURLE 2.0. The same process can be replicated for other adaptive e-learning systems that provide great adaptation and yet have limited usage because of their architectures and implementations.

The research community seems to agree that the next step of integrating Adaptive and Intelligent Web-based Educational Systems (AIES) will be achieved by defining learning frameworks or platforms that allow the creation of intelligent systems. Those platforms and frameworks should focus on being domain independent, extensible, interoperable and reusable technically and semantically (Trella *et al.* 2005). WHURLE 2.0 was designed to satisfy these requirements.

By addressing Adaptive Educational Hypermedia's issues, such adaptive Web-based systems will start to take their well-deserved positions in e-learning and penetrate into mainstream education.

BIBLIOGRAPHY

AUSTRAILIAN NATIONAL TRAINING AUTHORITY (2003). *Potential Benefits and Challenges in the Use of SCORM in VET* [WWW] VET LEARNING REPOSITORY PROJECT. Available from: http://pre2005.flexiblelearning.net.au/projects/resources/VLOR_potential_SCORM_benefits_challenges_for_VET.pdf [Retrieved: January 10, 2008].

ABDULLAH, N. A. and DAVIS, H. (2003). Is simple sequencing simple adaptive hypermedia? In: *Proceedings of the 14th ACM conference on Hypertext and hypermedia*, *Nottingham, UK*. ACM.

ABDULLAH, N. A., BAILEY, C. P. and DAVIS, H. C. (2004). Augmenting SCORM Manifests with Adaptive Links. In: *Proceedings of the15th ACM Conference on Hypertext and Hypermedia*, *Santa Cruz, USA*.

ABDULLAH, N. A. and DAVIS, H. C. (2005). A Real-time Personalization Service for SCORM. In: *Proceedings of the 5th IEEE International Conference on Advanced Learning Technologies (ICALT 2005)*, *Kaohsiung, Taiwan*, *July 5 - 8*.

ABERDOUR, M. (2007). *Open Source Learning Management Systems* [WWW] An Epic White paper, Epic. Available from: http://www.epic.co.uk/content/news/oct_07/whitepaper.pdf [Retrieved: January 13, 2008].

ABU-SHAWAR, B., AL-SADI, J. and HOURANI, A. (2006). Integrating the Learning Management System with other Online Administrative Systems at AOU. In: *Proceedings of the 2006 World Congress in Computer Science, Computer Engineering, and Applied Computing*, *Las Vegas, USA*, *26-29 June*.

ADISEN, A. and BARKER, T. (2007). Supporting the diversity of the E-learning 2.0 learners: The development of a psychological student model. In: *Proceedings of World Conference on E-Learning in Corporate, Government, Health Care, & Higher Education (e-Learn 2007)*, *Quebec City, Quebec, Canada*, *October 15-19*. AACOE.

Adkins, S. (2005). *Wake-Up Call: Open Source LMS* [WWW]. Available from: http://www.astd.org/LC/2005/1005_adkins.htm [Retrieved: July 23, 2008].

ADL. (2004). *SCORM 2004 3rd Edition* [WWW] ADVANCED DISTRIBUTED LEARNING. Available from: http://www.adlnet.gov/scorm/. [Retrieved: July 8, 2008].

ALEXANDER, B. (2006). Web 2.0: A New Wave of Innovation for Teaching and Learning? *EDUCAUSE Review* 41 (2), pp. 32– 44.

ARAPI, P., MOUMOUTZIS, N. and CHRISTODOULAKIS, S. (2003). Supporting Interoperability in an Existing E-learning Platform with SCORM. In: *Proceedings of* 3rd *IEEE International Conference on Advanced Learning Technologies (IEEE ICALT 2003)*, *Athens, Greece*, *July 09 – 11*.

ATUTOR. (2008). *ATutor*. Available from: http://www.atutor.ca/. [Retrieved: January 28, 2008].

AVGERIOU, P., PAPASALOUROS, A., RETALIS, S. and SKORDALAKIS, M. (2003). Towards a Pattern Language for Learning Management Systems. *Educational Technology & Society* 6 (2), pp. 11-24.

BAAJOUR, H., MAGOULAS, G.D. and POULOVASSILIS, A. (2007). Design Services-Enabled Personalised Support for Planning of Lifelong Learning Based on Individual and Group Characteristics. In: *Proceedings of Workshop on Personalisation in E-Learning Environments at Individual and Group Level at the 11th International Conference on User Modeling (UM 07)*, *Corfu, Greece*, *June 25-29*.

BEAL, V.A. (2005). *Understanding Web Services* [WWW] Webopedia. Available from: http://www.webopedia.com/DidYouKnow/Computer_Science/2005/web_services.asp. [Retrieved: 11 October 2005].

BERNERS-LEE, T., HENDLER, J. and LASSILA, O. (2001). The Semantic Web: A New Form of Web Content That Is Meaningful to Computers Will Unleash a Revolution of New Possibilities. *Scientific American Magazine* 284 (5), pp. 28-37.

BOYLE, C. and ENCARNACION, A.O. (1994). Metadoc: An adaptive hypertext reading system. *User Modeling and User-Adapted Interaction* 4 (1), pp. 1-19.

BRAILSFORD, T. J., MOORE, A., STEWART, C. D., ZAKARIA, M. R., CHOO, B. S. and DAVIES, P. C. (2001). Towards a Framework for Effective Web-based Distributed Learning. In: *Proceedings of 10th International World Wide Web Conference*, *Hong Kong*, *May 1-5*.

BRAILSFORD, T. J., ASHMAN, H., STEWART, C., ZAKARIA, M.R. and MOORE, A. (2002a). User Control of Adaptation in an Automated Web-Based Learning Environment. In: *Proceedings of the 1st International Conference on Information Technology & Applications (ICITA 2002)*, *Bathurst, Australia November 25-28*.

BRAILSFORD, T. J., STEWART, C.D., ZAKARIA, M.R. AND MOORE, A. (2002b). Autonavigation, Links and Narrative in an Adaptive Web-Based Integrated Learning Environment In: *Proceedings of 11th International Conference on the World Wide Web (WWW2002)*, *Honolulu, Hawaii (USA)*, *May 7 -11*.

BRICKLEY, D. AND MILLER, L. (2007). *FOAF Vocabulary Specification 0.91*[WWW]. Available from: http://xmlns.com/foaf/spec/.
[Retrieved: July 7, 2008].

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

BRUSILOVSKY, P. and PESIN, L. (1998). Adaptive navigation support in educational hypermedia: An evaluation of the ISIS-Tutor. *Computing and Information Technology* 6 (1), pp. 27-38.

BRUSILOVSKY, P., DE BRA, P. and HOUBEN, G. (1999). Adaptive Hypermedia: From System to Framework. *ACM Computing Surveys* 31 (4es).

BRUSILOVSKY, P. (2001a). Adaptive Educational Hypermedia. In: *Proceedings of Tenth International PEG conference*, *Tampere, Finland*, *June 23 - 26*.8-12.

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

BRUSILOVSKY, P. (2003a). A Distributed Architecture for Adaptive and Intelligent Learning Management Systems. In: *Proceedings of Workshop on Intelligent Learning Management Systems at AI-ED'2003 (11th International Conference on Artificial Intelligence in Education)*, *Sydney Australia 21, July, 2003*.106-114.

BRUSILOVSKY, P. (2003b). Developing adaptive educational hypermedia systems: From design models to authoring tools. In: T. MURRAY, B. BLESSING and S. AINSWORTH, eds. *Authoring Tools for Advanced Technology Learning Environment*. Dordrecht: Kluwer Academic Publishers: pp. 377- 409.

BRUSILOVSKY, P. (2004a). Adaptive Hypermedia: From Generation to Generation. In: *Proceedings of 4th Hellenic Conference on Information and Communication Technologies in Education*, *Athens, Greece*, *September 29 - October 3, 2004*. pp.19-33.

Brusilovsky, P. (2004b). KnowledgeTree: A distributed architecture for adaptive E-Learning In: *Proceedings of 13th International World Wide Web Conference, WWW 2004*, *New York, USA*, *May 17-22*. ACM Press, pp. 104 -113.

BRUSILOVSKY, P., SOSNOVSKY, S. and YUDELSON, M. (2005). Ontology-based Framework for User Model Interoperability in Distributed Learning Environments. In: G. Richards, ed *Proceedings of World Conference on E-Learning (E-Learn 2005)*, *Vancouver, Canada*, *October 24-28*. AACE, pp. 2851-2855.

BRUSILOVSKY, P. and HENZE, N. (2007). Open Corpus Adaptive Educational Hypermedia. In: P. BRUSILOVSKY, A. KOBSA, and W. NEIDL, eds. *The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science*. Berlin: Springer-Verlag, 4321, pp. 671-696.

BRUSILOVSKY, P., KOBSA, A. AND NEJDL, W. (eds). (2007). *The Adaptive Web: Methods and Strategies of Web Personalization* LNCS Berlin: Springer-Verlag.

BBRUSILOVSKY, P., WADE, V. AND CONLAN, O (2008). From Learning Objects to Adaptive Content Services for E-Learning. In: P. PAHL, Ed. *Architecture Solutions for E-Learning Systems*. Hershey, PA: Idea Group Inc, pp. 243-261.

Chan, T. (2007). Implementing Web 2.0 Technology for E-Learning Systems. In: T. BASTIAENS AND S. CARLINER, eds *Proceedings of World Conference on E-Learning in Corporate, Government, Health Care, & Higher Education (e-Learn 2007)*, *Quebec City, Quebec, Canada*, *15-19 October.* AACOE.

CHATTERJEE, S., and WEBBER, JAMES. (2004). *Developing Enterprise Web Services An Architect's Guide*. New Jersey: Prentice Hall.

COLE, J. AND FOSTER, H. (2005). *Using Moodle: Teaching with the Popular Open Source Course*: O'Reilly Media.

CONLAN, O., WADE, V., BRUEN, C. and GARGAN, M. (2002a). Multi-model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning. In: *Proceedings of The Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems London, UK*, *2002*. Berlin: Springer-Verlag 100-111.

CONLAN, O., WADE, V., GARGAN, M., HOCKEMEYER, C. AND ALBERT, D (2002b). An Architecture for integrating Adaptive Hypermedia Services with Open Learning Environments In: *Proceedings of ED-MEDIA 2002, World Conference on Educational Multimedia, Hypermedia & Telecommunications*, *Denver, Colorado*, *June 2002*.

CONLAN, O. (2003). State of Art: Adaptive Hypermedia. *State of Art Surveys* (2), pp. 47-136.

CONLAN, O. (2004a). *The Multi-model, Metadata-driven Approach to Content and Layout Adaptation* [PowerPoint Presentation]. Available from: www.w3.org/2004/06/DI-MCA-WS/presentations/originals/tcd.ppt [Retrieved: August 25, 2008].

CONLAN, O. and WADE, V. (2004b). Evaluation of APeLS - An Adaptive eLearning Service based on the Multi-model, Metadata-driven Approach. In: *Proceedings of 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based*

Systems (AH2004), *Eindhoven, The Netherlands*, *August 2004*. Berlin: Springer-Verlag, pp. 291 - 295.

CORCOLES, C., CASADO, C., HUERTAS, M.A, MOR, E. and GUERRERO-ROLDAN, A (2007). Learning 2.0: concepts and experiences with social networks software. In: T. BASTIAENS AND S. CARLINER *Proceedings of World Conference on E-Learning in Corporate, Government, Health Care, & Higher Education (e-Learn 2007)*, *Quebec City, Quebec, Canada*, *15-19, October 2007*. AACOE.

CRISTEA, A. and DE MOOIJ, A. (2003a). LAOS: Layered WWW AHS Authoring Model and its corresponding Algebraic Operators. In: *Proceedings of 14th International World Wide Web Conference*, *Budapest, Hungary May 20-24* ACM.

CRISTEA, A. I. and DE MOOIJ., A. (2003b). Adaptive Course Authoring: My Online Teacher. In: *Proceedings of ICT'03*, *Papeete, French Polynesia.*

CRISTEA, A., SMITS, D. and BRA, P (2005). Writing MOT, reading AHA! : converting between an authoring and a delivery system for adaptive educational hypermedia In: A.I. CRISTEA, R. CARRO and F. GARZOTTO, eds *Proceedings of 3rd International Workshop on Authoring of Adaptive and Adaptable Educational Hypermedia (A3EH; in conjunction with AIED'05)*, *Amsterdam, The Netherlands*, *July 18-22*.

CRISTEA, A., STEWART, C., BRAILSFORD, T. and CRISTEA, P. (2005). Evaluation of Interoperability of Adaptive Hypermedia Systems: testing the MOT to WHURLE conversion in a classroom setting. In: *Proceedings of 3rd Adaptive and Adaptable Educational Hypermedia workshop at the Artificial Intelligence in Education 2005*, *Amsterdam, The Netherlands*, *July 18-22, 2005*.

DAGGER, D., O'CONNOR, A., LAWLESS, S., WALSH, E. and WADE, V. (2007) Service-Oriented E-Learning Platforms: From Monolithic Systems to Flexible Services. *IEEE Internet Computing*, May-June, pp. 28-35.

DE BRA, P., HOUBEN, G-J. and WU, H. (1999). AHAM: a Dexter-based reference model for adaptive hypermedia. In: *Proceedings of* 10th *ACM Conference on Hypertext and hypermedia: Returning to Our Diverse Roots*, *Darmstadt, Germany*. ACM.

DE BRA, P. (2000). Pros and Cons of Adaptive Hypermedia in Web-based Education. *CyberPsychology & Behavior.* 3 (1), pp. 71-77.

DE BRA, P. (2002). Adaptive Educational Hypermedia on the Web. *Communication of the ACM* 45 (5), pp. 60-61.

DE BRA, P., AERTS, A., BERDEN, B., DE LANGE, B., ROUSSEAU, B., SANTIC, T., SMITS, D. and STASH, N. (2003). AHA! The Adaptive Hypermedia Architecture. In: *Proceedings of ACM Hypertext Conference*, *Nottingham, UK*, *26-30 August 2003*.81-84.

DE BRA, P., AROYO, L. and CHEPEGIN, V. (2004). The Next Big Thing: Adaptive Web-Based Systems. *Journal of Digital Information* 5 (1), pp. Article No.247.

DE JONG, I. (2002). *Web Services/SOAP and CORBA* [WWW]. Available from: http://www.xs4all.nl/~irmen/comp/CORBA_vs_SOAP.html#4 [Retrieved: August 8, 2008].

DILLENBOURGH, P. (1999). What do you mean by 'collaborative learning'? In: P. DILLENBOURGH, ed. *Collaborative Learning: Cognitive and Computational Approaches. Advances in Learning and Instruction Series.* Oxford: Elsevier Science, pp. 1-19.

DODDS, L. (2005). Connecting Social Content Services using FOAF, RDF and REST. In: *Proceedings of XTech 2005: XML, the Web and beyond*, *Amsterdam, Netherlands*, *May 24-27*.

DOLOG, P., WOLFGANG, N., HENZE, N. and SINTEK, M. (2004). Personalization in Distributed e-Learning Environments. In: *Proceedings of* Proceedings *of the 13th international World Wide Web conference New York, NY, USA*, *May 17-22* ACM Press.

DOWNES, S. (2005). E-learning 2.0. *eLearn Magazine, ACM*.

DUFFY, P. and BRUNS, A. (2006). The use of blogs, wikis and RSS in education: A conversation of possibilities. In: *Proceedings of The Online Learning and Teaching Conference 2006*, *Brisbane, Australia*, *September 26*.

EKLUND, J. and BRUSILOVSKY, P. (1998). The Value of Adaptivity in Hypermedia Learning Environments: A Short Review of Empirical Evidence. In: *Proceedings of 2nd Workshop on Adaptive Hypertext and Hypermedia at the Ninth ACM International Hypertext Conference: Links, Objects, Time and Space - Structure in Hypertext'98*, *Pittsburgh, PA, USA*, *June 20-24*. ACM.

FELDER, R. M. AND SOLOMAN, B. A. *Index of Learning Styles* [WWW]. Available from: http://www.ncsu.edu/felder-public/ILSpage.html [Retrieved: 1st August, 2008].

FELDER, R. M. and SILVERMAN, L.K. (1988). Learning and Teaching Styles in Engineering Education. *Engineering Education* 78 (7), pp. 674-681.

FIELDING, R. (2000) *Architectural Styles and the Design of Network-based Software Architectures, Chapter (3): Representational State Transfer (REST).* Doctoral thesis (PhD), University Of California.

FIELDING, R. and TAYLOR, N. (2000). Principled design of the modern Web architecture. In: *Proceedings of 22nd International Conference on Software Engineering*, *Limerick, Ireland*. ACM.

FRIESEN, K. and MAZLOUMI, N. (2003). Transparent Integration of Learning Management Systems and Third-party Applications Using Web Services In: V. USKOV, ed *Proceedings of Computers and Advanced Technology in Education (CATE 2003)*, *Rhodes, Greece*, *June 30 - July 2*.

FUMERO, A. (2006). EDUWEB 2.0 - iCamp & N-Gen Educational Web. In: *Proceedings of Proceedings of WEBIST conference*, *Setubal, Portugal*, *April 11-13*.

GRAF, S. (2005a). Fostering Adaptivity in E-Learning Platforms: A Meta-Model Supporting Adaptive Courses. In: *Proceedings of IADIS International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2005) Porto, Portugal.*

GRAF, S. AND LIST, B. (2005b). An Evaluation of Open Source E-Learning Platforms Stressing Adaptation Issues.

GRAF, S. (2007). Providing Adaptive Courses in Learning Management Systems with Respect to Learning Styles. In: *Proceedings of* e-*Learn 2007*, *Quebec City, Quebec, Canada AACOE*, *15 - 19 October, 2007*. AACOE.

GRAPPLE. (2008). *What is GRAPPLE ?* [WWW]. Available from: http://www.grapple-project.org/. [Retrieved: August 8, 2008].

GREW, P., PAGANI, E. and GIUDICI, F. (2008). From e-Learning to M-learning: Architectures to Support University Teaching. In: C. PAHL, ed. *Architecture Solutions For E-Learning Systems*. Hershey, USA: Idea Group Inc., pp. 62-79.

GROSSMAN, L. (2006) Time's Person of the Year: You. *Time Magazine*, Wednesday, December. 13, 2006.

ALONSO, G., CASATI, F., KUNO, H. and MACHIRAJU, V. (2004). *Web Services: Concepts, Architecture and Application*. Berlin Heidelberg: Springer-Verlag.

GÜTL, C. (2008). Moving Towards a Generic, Service-based Architecture for Flexible Teaching and Learning Activities. In: C. PAHL, ed. *Architecture Solutions for E-Learning Systems*. Hershey, USA.: Idea Group Inc., pp. 1-24.

HARRER, A. and MARTENS, A. (2008). A Step Towards a Pattern Language for E-Learning Systems. In: C. PAHL, ed. *Architecture Solutions For E-Learning Systems* Hershey, USA: Idea Group Inc., pp. 98-115.

HENZE, H. (2005). Personalization Services for the Semantic Web: The Personal Reader Framework, Framework 6 Project Collaboration. In: *Proceedings of Future Semantic Web Workshop at European Semantic Web Conference ESWC 2005*, *Heraklion, Greece*, *May 29 - June 1*.

HINCHCLIFFE, D. (2005). *The Hidden Battle Between Web Services: REST versus SOAP* [Weblog Online] Dion Hinchcliffe's Blog - Musings and Ruminations on Building Great Systems: Lightweight Processes, Service-Orientation, Enterprise Architecture, and Software Development. Available from: http://hinchcliffe.org/archive/2005/02/12/171.aspx [Retrieved: August 8, 2008].

HUSSEIN, S. (2005). *Developing e-Learning Materials: Applying User-Centred Design*. London: National Institute of Adult and Continuing Education.

IBM DEVELOPERWORKS. (2008). *Standards and Web Services* [WWW] IBM. Available from: http://www.ibm.com/developerworks/webservices/standards/. [Retrieved: August 8, 2008].

JEAN-LUC, D. (2004). *Creating and Consuming Web Services with PHP* [WWW]. Available from: http://webservices.xml.com/pub/a/ws/2004/03/24/phpws.html. [Retrieved: March 28, 2007].

KAY, J., KUMMERFELD, B. and LAUDER, P. (2002). Personis: A server for User Models In: P. BRUSILOVSKY, P. DE BRA and R. CONEJO, eds *Proceedings of Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002)*, *Malaga, Spain*, *29 - 31, May* 201-212.

KAYAMA, M. and NAGATA, N. (2008). Collaborative E-Learning Systems and E-Pedagogy: Learning Resources Infrastructure. In: C. PAHL, ed. *Architecture Solutions For E-Learning Systems*. Hershey, USA: Idea Group Inc., pp. 25-43.

KAYE, D. (2003). *Loosely Coupled: The Missing Pieces of Web Services*. Marin County, California: RDS Strategies LLC.

KEN NORTH COMPUTING. (2007). *Middleware in Action: Industrial Strength Data Access* [WWW] KEN NORTH COMPUTING LLC. Available from: http://www.sqlsummit.com/Articles/DataAccessMiddleware6.htm. [Retrieved: August 6, 2008].

LAFORCADE, P., NODENOT, T., CHOQUET, C. and CARON, P. (2008). Model-Driven Engineering (MDE) and Model-Driven Architecture (MDA) Applied to the Modelling and Development of Technology Enhanced Learning (TEL)

Systems: Promises, Challenges and Issues. In: C. PAHL, ed. *Architecture Solutions for E-Learning Systems*. Hershey, USA: Idea Group Inc, pp. 116-136.

LP, A. and CANAL, R. (2003). Supporting Collaborative Learning Activities with SCORM. *EDUCAUSE IN AUSTRALASIA* 3, pp. 669-678.

MAPFORCE. (2006). *Altova MapForce* [WWW]. Available from: http://www.altova.com/download_mapforce.html [Retrieved: March 30, 2006].

MARSHALL, C.C. and SHIPMAN, F. M. (2003). Which Semantic Web? In: *Proceedings of 14th ACM conference on Hypertext and Hypermedia*, *Nottingham, UK*. ACM.

METTE, K. and THOMAS, C. G. (1994). Adaptivity: system-initiated individualization. In: R. OPPERMANN, ed. *Adaptive user support: ergonomic design of manually and automatically adaptable software*. Hillsdale, NJ : Lawrence Erlbaum Associates, Inc., pp. 67-96.

MILLARD, D., DOODY, K., DAVIS, H., GILBERT, L., HOWARD, Y., TAO, F. and WILLS, G. (2008). (Semantic Web) Services for e-Learning. International Journal of Knowledge and Learning (IJKL) (4).

MILLARD, M. and ESSEX, C. (2007). Web 2.0 Technologies for Social and Collaborative E-Learning: We and Our Students are the Web. In: T. BASTIAENS AND S. CARLINER, eds *Proceedings of World Conference on E-Learning in Corporate, Government, Health Care, & Higher Education (e-Learn 2007)*, *Quebec City, Quebec, Canada 15-19, October 2007*. AACOE.

MOODLE. (2008). *Moodle* [WWW]. Available from: http://moodle.org [Retrieved: January 1, 2008].

MORENO, A., BEDNARIK, R. and YUDELSON, M. (2007). How to Adapt the Visualization of Programs? In: *Proceedings of* Workshop *on Personalisation in E-Learning Environments at Individual and Group Level at the 11th International Conference on User Modeling (UM 07)*, *Corfu, Greece*, *June 25-29*.

MYLONAKIS, N. and CULLOUGH, C.M. (2003). Adaptive Hypermedia, its role in lifelong learning. In: *Proceedings of* Policy*, Practice and Partnership: Getting to work on Lifelong Learning*, *Cedefob, Thessaloniki*, *2-3 June.*

NAGAPPAN, R., SKOCZYLAS, R. and SRIGANESH, R. P. (2003). *Developing Java Web Services: Architecting and Developing Secure Web Services Using Java.* Indianapolis, Indiana: Wiley Publishing.

NARAMORE, E., GERNER, J., LE SCOUARNEC, Y., STOLZ, J. and GLASS, M.K (2005). *Beginning PHP, Apache, MySQL Web Development*. Indianapolis, Indiana: Wiley Publishing.

NICHOL, S. (2004). *Introduction to NuSOAP* [WWW]. Available from: http://www.scottnichol.com/nusoapintro.htm [Retrieved: June 27, 2008].

O'REILLY, T. (2005). *What is Web 2.0?*[WWW] O'REILLY Media. Available from: http://www.oreilly.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html. [Retrieved: December 3, 2007].

O'REILLY, T. (2006). *Web 2.0: Compact definition* [WWW] O'REILLY RADAR. Available from: http://radar.oreilly.com/archives/2005/10/web_20_compact_definition.html. [Retrieved: December 3, 2007].

OBEIDAT, H. (2008) *Authoring Adaptive Educational Hypermedia to Multiple User Models*. Masters thesis (MSc. in Management of IT), University of Nottingham.

OLIVER, B. and HAWKER, A. (2005). *e-Framework programme* [WWW]. Available from: http://www.jisc.ac.uk/whatwedo/programmes/programme_eframework.aspx. [Retrieved: May 11, 2008].

OPPERMANN, R., ed. (1994). *Adaptive user support: ergonomic design of manually and automatically adaptable software*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

OUBAHSSI, L. and GRANDBASTIEN, M. (2008). E-Learning Systems Re-Engineering: Functional Specifications and Component-Based Architecture. In: C. PAHL, Ed. *Architecture Solutions For E-Learning. Systems*. Hershey, USA: Idea Group Inc., pp. 175-192.

OUNNAS, A., DAVIS, H. D. and D.E., M. (2007). Semantic Modeling for Group Formation. In: *Proceedings of Workshop on Personalisation in E-Learning Environments at Individual and Group Level at the 11th International Conference on User Modeling (UM 07), Corfu, Greece*.

PAPANIKOLAOU, K., GOULI, E. and GRIGORIADOU, M. (2007). Investigating Individual-to-Group and Group-to-Individual Influences. In: *Proceedings of Workshop on Personalisation in E-Learning Environments at Individual and Group Level at the 11th International Conference on User Modeling (UM 07), Corfu, Greece*.

PARAMYTHIS, A. and LOIDL-REISINGER, S. (2004). Adaptive Learning Environments and e-Learning Standards. *Electronic Journal of e-Learning (EJEL)* 2 (2).

PHP. (2008). *xslt_proces* [WWW]. Available from: http://uk.php.net/xslt_process. [Retrieved: 27-6-2008].

PLEZ, C. (2003) Web Services Orchestration and Choreography. *IEEE Computer*, October pp. 46-52.

POTTS, S. AND KOPACK, M. (2003). *SAMS Teach Yourself Web Services.* Indiana: SAMS.

POWELL, G. (2007). *Beginning XML Databases*. Indianapolis, Indiana: Wiley Publishing.

REY LÓPEZ, M., FERNÁNDEZ VILAS, A., DÍAZ REDONDO, R. P., PAZOS ARIAS, J. J. and BERMEJO MUÑOZ, J. (2006). Extending SCORM Too Create Adaptive Courses. *Lecture notes in computer science LNCS: Innovative Approaches for Learning and Knowledge Sharing (Proceedings of 1st. European Conference on Technology Enhanced Learning, EC-TEL 2006, Crete, Greece, October 1-4, 2006)* 4227 (679-684).

REY LÓPEZ, M., BRUSILOVSKY, P., MECCAWY, M., DÍAZ REDONDO, R., FERNÁNDEZ VILAS, A. AND ASHMAN, H. (2007). Resolving the Problem of Intelligent Learning Content in Learning Management Systems. *International Journal on E-Learning* 7 (3).

REY-LÓPEZ, M., DÍAZ-REDONDO, R. P., FERNÁNDEZ-VILAS, A., PAZOS-ARIAS, J. J., GARCÍA-DUQUE, J., GIL-SOLLA, A. and RAMOS-CABRER, M. (2008). An extension to the ADL SCORM standard to support adaptivity: The t-learning case-study. *Computer Standards & Interfaces*. In Press, corrected proof, to appear in print in 2009.

ROSEN, A. (2006). Technology Trends: e-Learning 2.0. *Learning Solutions e-Magazine* [WWW],October 9. Available from: http://www.readygo.com/e-learning-2.0.pdf [Retrieved: May 11, 2008].

SAFRAN, C., GUETL, C. and HELIC, D. (2007). The Impact of Web 2.0 on Learning at a Technical University - A usage survey. In: T. BASTIAENS and S. CARLINER, eds *Proceedings of E-Learn 2007*, *Quebec City, Quebec, Canada*, *15-19, October*. AACOE.

SAKAI. (2008). *SAKAI*. Available from: http://sakaiproject.org/portal. [Retrieved: September 8, 2008].

SANGINETO, E. (2008). An Adaptive E-Learning Platform for Personalized Course Generation. In: C. PAHL, ed. *Architecture Solutions for E-Learning Systems*. Hershey, USA: Idea Group Inc, pp. 262-281.

SANTOS, J., ANIDO, L., and LIAMAS, M. (2003). On the Use of E-Learning Standards in Adaptive Learning Systems. In: *Proceedings of 3rdIEEE International Conference on Advanced Learning Technologies (ICALT'03).*

SODOKE, K., RIOPEL, M., RAÎCHE, G., NKAMBOU, R. and LESAGE, M. (2007). Extending Moodle Functionalities to Adaptive Testing Framework. In: G. Richards, ed *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (e-learn 2007) Quebec City, Quebec, Canada 15-19 October, 2007*. AACE. pp. 476-482.

SOTOMAYOR, B. (2005). *A short introduction to Web Services* [WWW] The Globus Toolkit 4 Programmer's Tutorial. Available from: http://gdp.globus.org/gt4-tutorial/multiplehtml/ch01s02.html [Retrieved: August 7, 2008].

ST. LAURENT, S., JOHNSTON, J. and DUMBILL, E. (2001). *Programming Web Services with XML-RPC*. Sebastopol: O'Reilly & Associates, Inc.

STASH, N., CRISTEA, A. and DE BRA, P. (2006). Adaptation to Learning Styles in E-Learning: Approach Evaluation. In: *Proceedings of E-Learn 2006 Honolulu, Hawaii*, *October 2006.*

STEWART, C. (2003). *A Very Quick Introduction to WHURLE* [PowerPoint Presentation]. Available from: *wwwis.win.tue.nl/~acristea/AH-Ro/AH3.ppt* [Retrieved: August 2, 2008].

STEWART, C., CRISTEA, A., MOORE, A., BRAILSFORD, T. and ASHMAN, H. (2004). Authoring and Delivering Adaptive Courseware. In: *Proceedings of* International *Workshop on Authoring of Adaptive and Adaptable Educational Hypermedia*.

STEWART, C., CRISTEA, A., BRAILSFORD, T. and ASHMAN, H. (2005). Authoring Once, Delivering Many: Creating Reusable Adaptive Courseware. In: *Proceedings of IASTED International Conference on Web-based Education (WBE '05)*, *Grindelwald, Switzerland*, *February 21-23, 2005.*

STEWART, C., CELIK, I., CRISTEA, A. and ASHMAN, H. (2006). Interoperability between AEH user models. In: *Proceedings of joint international workshop on Adaptivity, personalization & the semantic web*, *Odense, Denmark* ACM Press.21-30.

STRATEGIC PLANNING ALIGNEDSTRATEGY. (2006) *IT Strategic Plan: Process Innovation* [WWW]. Available from:

http://www.alignedstrategy.com/weblog/labels/architecture.htm.4/8/2008. [Retrieved: August 9, 2008]

SUBBIAN, K. and KANNAN, R. (2006). *Enable C++ applications for Web service using XML-RPC* [WWW] IBM. Available from: http://www.ibm.com/developerworks/webservices/library/ws-xml-rpc/. [Retrieved: July 15, 2008].

SUN, L. and FU, Y. (2005). Interoperability For E-learning Services Management and Provision. *World Wide Web,* 8 (4), pp. 395-412.

TECHREPUBLIC. (2002). *SCORM: A sharable and interoperable model for e-learning* [WWW]. Available from: http://articles.techrepublic.com.com/5100-10878_11-1045417.html [Retrieved: September 23, 2008].

TIBCO Developer Network. (2008) *Service Oriented Client* [WWW]. Available from http://www.tibco.com/devnet/[Retrieved: August 4, 2008].

TRELLA, M., CARMONA, C. and CONEJO, R. (2005). MEDEA: an Open Service-Based Learning Platform for Developing Intelligent Educational Systems for the Web. In: P. BRUSILOVSKY, R. CONEJO and E. MILLÁN, eds *Proceedings of Workshop on Adaptive Systems for Web-Based Education: Tools and Reusability at the 12th International Conference on Artificial Intelligence in Education (AIED 2005)*, *Amsterdam, The Netherlands*.27-34.

TRURAN, M., GOULDING, J. and ASHMAN, H. (2005). Co-active intelligence for image retrieval. In: *Proceedings of 13th Annual ACM international Conference on Multimedia (MULTIMEDIA '05)*, *Singapore*, *November 06 - 11, 2005.* ACM Press. pp. 547-550.

TU, C.-H. (2000). On-line learning migration: from social learning theory to social presence theory in a CMC environment. *Journal of Network and Computer Applications* 23 (1), pp. 27-37.

TYAGI, S. (2006). *RESTful Web Services* [WWW] Java's Sun Microsystems Technical Articles Available from: http://java.sun.com/developer/technicalArticles/WebServices/restful/. Retrieved: [Retrieved: April 16, 2008]

VASILYEVA, E., PECHENIZKIY, M. and DE BRA, P. (2007). Adaptation of Feedback in e-learning System at Individual and Group Level. In: *Proceedings of Workshop on Personalisation in E-Learning Environments at Individual and Group Level at the 11th International Conference on User Modeling (UM 07)*, *Corfu, Greece*, *June 25-29*.

VERGARA EDE, M. and CAPUANO, N. (2003). DIOGENE: A Training Web Broker for ICT Professionals. In: *Proceedings of* 2nd *International Conference on Multimedia and ICT in Education (m-ICTE 2003)*, *Badajoz, Spain.* pp. 330-334.

W3C. (2001). *Web Services Description Language (WSDL) 1.1* [WWW] The World Wide Web Consortium. Available from: http://www.w3.org/TR/wsdl [Retrieved: June 28, 2008].

W3C. (2002a). *How the Semantic Web Works* [WWW] The World Wide Web Consortium. Available from: http://www.w3.org/2002/03/semweb/ [Retrieved: May 20, 2008].

W3C. (2002b). *Web Service Choreography Interface (WSCI) 1.0* [WWW] The World Wide Web Consortium. Available from: http://www.w3.org/TR/wsci/ [Retrieved: August 1, 2008].

W3G. (2004). *Web Services Glossary*. [WWW] The World Wide Web Consortium. Available from: http://www.w3.org/TR/ws-gloss/ [Retrieved: June 28, 2008].

W3C. (2006). *SOAP Version 1.2 Part 0: Primer (Second Edition)* [WWW] The World Wide Web Consortium. Available from: http://www.w3.org/2000/xp/Group/2/06/LC/soap12-part0.html [Retrieved: August 8, 2008].

W3C. (2007). *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)* [WWW] The World Wide Web Consortium. Available from: http://www.w3.org/TR/2007/REC-soap12-part1-20070427/ [Retrieved: September 20, 2008].

WALCZOWSKI, L. and ELLIS., M. (2007). A Learning Management Portal for E-Learning 2.0: a Teacher's and Developer's Prospective. In: T. BASTIAENS and S. CARLINER, eds *Proceedings of E-Learn 2007*, *Quebec City, Quebec, Canada*, *15-19, October 2007*. AACOE.

WEBER, G., KUHL, C.-H. and WEIBELZAHL, S. (2002). Developing Adaptive Internet Based Courses with the Authoring System NetCoach. In: *Proceedings of Revised Papers from the International Workshops OHS-7, SC-3, and AH-3 on Hypermedia: Openness, Structural Awareness, and Adaptivity*. Berlin: Springer-Verlag.

WEBOPEDIA. (2001). *Distributed Computing* [Online Technical Dictionary]. Available from: http://www.webopedia.com/TERM/D/distributed_computing.html [Retrieved: August 1, 2008].

WEBOPEDIA. (2006). *Service-Oriented Architecture* [Online Technical Dictionary]. Available from:

http://www.webopedia.com/TERM/S/Service_Oriented_Architecture.html [Retrieved: August 2, 2008].

WELLER, M. (2006). VLE 2.0 and future directions in the learning environments. In: *Proceedings of 1st LAMS Conference*, *Sydney*, *December 2006*.

WRIGHT, M. (2005) *A Detailed Investigation Of Interoperability For Web Services*. Masters thesis (MSc), Rhodes University.

XML-RPC. (2003). *What is XML-RPC?* [WWW]. Available from: http://www.xmlrpc.com/[Retrieved: August 7, 2008].

YANK, K. (2001). Transform your PHP with XSLT [WWW] Site Point. Available from: http://www.sitepoint.com/article/transform-php-xslt/ [Retrieved: August 4, 2008]

ZAKARIA, M. R., MOORE, A., ASHMAN, H., STEWART, C.D. and BRAILSFORD, T. (2002). The Hybrid Model for Adaptive Educational Hypermedia. In: *Proceedings of second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002)*, *Malaga, Spain*, *29 - 31, May 2002*.580-585.

ZAKARIA, M. R., MOORE, A., STEWART, C.D. and BRAILSFORD, T.J. (2003). Pluggable User Models for Adaptive Hypermedia in Education. In: *Proceedings of ACM Hypertext Conference*, *Nottingham, UK*, *26-30 August 2003*.

zUR MUEHLEN, M., NICKERSON, J. V. and SWENSON, K. D. (2005). Developing web services choreography standards: the case of REST vs. SOAP. *Decision Support Systems.* 40 (1), pp. 9-29.

APPENDICES

<u>APPENDIX-A</u>: Moodle's Client

```
<?php
//This client is responsible for the adaption content extraction
//and loading into the tables.
require_once('nusoap.php');
require_once('../../config.php');
require_once('../../lib/datalib.php');
require_once($CFG->dirroot.'/mod/lesson/lib.php');
require_once($CFG->dirroot.'/mod/lesson/locallib.php');

$username=$USER->username;

//To get the Lesson's object
   $id      = required_param('id', PARAM_INT);         // Course Module ID
   list($cm, $course, $lesson) = lesson_get_basics($id);
   $lessonname=$lesson->name;

  $wsdl="http://bubba.cs.nott.ac.uk/AGS/AGS.php?wsdl";
  $client=new soapclient($wsdl, 'wsdl');
  $param=array("username"=>$username, "coursename"=>$lessonname);
  $result=$client->call('AGS', $param);

//pushing results into mdl_lesson_pages table
  $dbname = "mzm";
  $dbuser = "";
  $dbpass = "";
  $dbserver = "localhost";
  $tablename = "mdl_lesson_pages";

$conn = mysql_connect($dbserver, $dbuser, $dbpass) or
die('Error connecting to mysql');
mysql_select_db($dbname);
$sql_query ="UPDATE $tablename SET contents='" .
mysql_real_escape_string($result)
."' WHERE lessonid='$lesson->id'";
$c_page = mysql_query($sql_query, $conn) or
die('counter INSERT error: '.mysql_errno().', '.mysql_error());
mysql_close($conn);
?>
```

<u>APPENDIX-B</u>: ATutor's Client

```
<?php
 //This client is responsible for the adaption content extraction
 //and loading into the tables

  define('AT_INCLUDE_PATH', 'include/');
  require(AT_INCLUDE_PATH . 'vitals.inc.php');
  require_once('include/classes/nusoap.php');

 //First parameter:
  $username=$_SESSION[login];

 //second parameter
  $lessonname='Use Case Lesson';

 //calling service
  $wsdl="http://bubba.cs.nott.ac.uk/AGS/AGS.php?wsdl";
  $client=new soapclient($wsdl, 'wsdl');
  $param=array("username"=>$username, "coursename"=>$lessonname);
  $result=$client->call('AGS', $param);

 //pushing results into AT_ content table
  $dbname = "atutor";
  $dbuser = "";
  $dbpass = "";
  $dbserver = "localhost";
  $tablename = "AT_content";

 $conn = mysql_connect($dbserver, $dbuser, $dbpass) or
  die('Error connecting to mysql');

  mysql_select_db($dbname);
  $sql_query ="UPDATE $tablename SET text='" .
mysql_real_escape_string($result)
. "' WHERE content_id='8'";
  $c_page = mysql_query($sql_query, $conn) or
  die('counter INSERT error: '.mysql_errno().', '.mysql_error());
  mysql_close($conn);
 ?>
```