USABILITY IN USER GENERATED LEARNING SPACES

A dissertation submitted to The University of Manchester

for the degree of Master of Science

in the Faculty of Engineering and Physical Sciences

2011

By

Blessing Sunday Udoisang

School of Computer Science

Table of Contents

Word Count: 36136

List of Acronyms

CPD	Continual Professional Development
PLE	Personal Learning Environment
PDP	Personal Development Planning (Planner, Plan)
PSC	Private Shareable Component
FLV	Flash Video
AAC	Advanced Audio Coding
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
PPMCC	The Pearson Product Moment Correlation Coefficient

List of Tables

List of Figures

Abstract

The focus of this work is to help enable learning in shared, collaborative multi-media learning spaces, by improving the facilities of, and the user interface to, the Learning Spaces within the Manchester Personal Learning Environment (the PLE).

The PLE and its learning spaces are designed to support learning as a social process: People learn with each other and from each other. The effective design of learning spaces can enhance the way learning takes place and consequently the outcomes of learning.

Established theories of learning give rise to strategies for learning facilitation. Behaviourism, Cognitivism, Social Constructivism, Papert's Constructionism, and Personal Development Planning are surveyed as an initial step in seeking informants for the work. Particularly, the view is taken that for present day learners who in part exist in a Web ecosystem, learning is all about knowledge creation and discovery, sharing and reusing content. The findings from the literature are used as a basis to improve usability in the PLE's virtual learning spaces. An improved and extended implementation is provided and evaluated, leading to suggestions for further improvements.

The results of the project indicate that usable virtual learning spaces can be constructed to support active, participatory and collaborative learning. The results also indicate that the amount of background computer and internet use by users has little or no bearing on how usable they find a virtual learning space.

Declaration

No portion of the work referred to in this dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Intellectual Property Statement

i. The author of this dissertation (including any appendices and/or schedules to this dissertation) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this dissertation, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has entered into. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the dissertation, for example graphs and tables ("Reproductions"), which may be described in this dissertation, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this dissertation, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/display.aspx?DocID=487), in any relevant Dissertation restriction declarations deposited in the University Library, The University Library's regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University's Guidance for the Presentation of Dissertations.

Acknowledgements

I hereby express my sincere appreciation to my supervisor, Mark van Harmelen (Dr.), for his priceless support during the project and the preparation of this dissertation.

Thank you very much David Workman, the '*Techie Genius*' of Hedtek for your invaluable technical support all through the project.

I also express my thanks to Tim Morris (Dr.) for his support throughout the period of the background research for this project.

Many thanks also to those who participated in the user evaluation.

Dedication

A dedication to Africa; the future is bright, but it begins now.

To Mr & Mrs Udoisang and all the little Isangs, I love you all.

To PTDF, Nigeria; I have achieved the aim of my coming, thank you.

To Bolanle Solomon, Bilkisu Bello, Abioye PLC, Kenny Popoola, Tunde Agbeja, Oyeniyi Oladeji, Okpobia Suo, Basil Pablo, Ayo-mi Adeku, Deborah Lartey; imagine Manchester without you.

Chapter 1

Introduction

Learning spaces “encompass the full range of places in which learning occurs, from real to virtual; classroom to chat room” (Brown, 2005). The effective design of physical and virtual learning spaces can enhance the way learning takes place and consequently the outcome. With the advancements in technology leading to convergence of platforms and proliferation of high capacity mobile devices, learners are increasingly shifting their preferences for learning environments from the physical to the virtual. For any interactive system that implements virtual learning spaces, usability is a key determinant in uptake. Usability is defined by the International Organization for Standardization (ISO 9241) as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use".

Designing for learning requires an understanding of how learners construct knowledge. Relevant learning theories need to be explored and applied within the learners' context to capture the learning activity in a way "natural" to the learners. Established theories such as social constructivism, constructionism, self-directed learning, communities of practice, etc engender strategies for facilitating learning. However, the context of learning keeps evolving. It is therefore important to re-examine these theories with a view to adapting them to the prevailing contexts in which learning occurs. This becomes more challenging when viewed from the virtual learning perspective. Interestingly, information and communication technology (ICT) provides a lot of tools that can help us realise our objectives. These ICT tools have been applied in various ways but the question is how usable are they?

This project is concerned with improving usability in user generated learning spaces. User generated learning spaces are “(learning) spaces which are populated with content by one or more learners” (van Harmelen, 2011). This content according to van Harmelen “might be learner generated, or co-opted from elsewhere and used unchanged, or modified, or mixed with other content”. Whatever the case might be, the purpose remains the same, “to help the user(s) learn about a particular topic area, or fulfil one or more learning goals” (van Harmelen, 2011). The project is motivated by the need to support learning among diverse categories of learners. Doing this raises vital questions which are clearly discussed in the next section.

1.1 Research Motivation and Questions

Learning is a continuous process; one does not stop learning (Dechant, 1991, Doukidis, Mylonopoulos & Pouloudi, 2004, Bergan, 2007). However, there are stages in our life when learning is formalized such that one is more bounded by rules (rather than choice) as to what to learn, when to learn it, how to show that what is required has been learnt and a reward (or otherwise) given for successfully showing it. Apart from this kind of formal system which majorly happens in academic institutions, one can also learn on his own. This kind of learning is referred to as personal learning, personal study, personal development, etc. As an example, a graduate lawyer who majored generally in oil and gas law might become particularly interested in oil financing laws. He[1] does not have to return to law school to get a degree for such knowledge; rather, he takes up a personal study and gets acquainted with the practice of oil financing laws. This kind of scenario is more formally embodied in what is known as Continuing Professional Development (CPD) (Rughani, Franklin & Dixon, 2003). However, it must not be formal to produce results if properly supported. The challenge is that such kind of personal learning is usually not very effective in the absence of a formal learning framework. This is because there is no external authority mandating the learner to carry out such learning effectively. Nonetheless, the serious learner, for personal reasons will proceed to carry out his personal studies and with the availability of a proper supporting framework, will do it effectively thus reaping the expected outcomes. Such supporting framework can be in form of formal/informal processes, coaching, mentoring, group studies and of course technology. This is where Personal Learning Environments (PLE) comes to play. A personal learning environment can support the learner to achieve his learning goals in many ways among which are:

- It can act as a repository of knowledge for the learner
- It can enable the learner to store and bookmark learning resources
- It can provide a social layer where the learner can collaboratively share knowledge with other learners
- It can help the learner monitor progress of his learning

[1] Masculine pronouns in this dissertation are taken to refer to any gender except where it is explicitly stated otherwise.

- It can help the learner reflect upon his learning and evaluate the outcome of such learning.

The motivation behind this research project is to support the learner in the ways listed above and in many more ways to achieve his learning goals. In order achieve this, we focus on usability in user generated learning spaces which has already been briefly discussed in the introduction and is further discussed in section 2.6.3. The project focuses on how to use interactive virtual learning spaces to support the user in pursuing his personal development in the presence or absence of a formal learning framework. Although there are various interactive multimedia tools in use today for similar purposes, this project is unique in that it does not focus on interactive multimedia technology in isolation; rather, it attempts to lay solid psychological and pedagogical foundations for the tools, processes and methods provided. Consequently, the research seeks to answer the following vital questions:

- Can one use interactive multimedia spaces to actively support a learner in achieving his/her learning goals?
- Can one develop a usable multimedia learning space that will run smoothly despite the limitations of the internet?
- Can the usability of such multimedia learning space be evaluated by users?
- Can a learner develop a personal development plan (PDP) in the multimedia learning space and use it to pursue a learning goal successfully?
- Can a personal development plan so created by a learner be converted to a learning artefact which can be used and/or re-used by other learners to achieve similar outcome(s) as the initial creator/learner?
- Can a learner share his personal development plans and pursue his learning goals collaboratively?
- Can a learner recover components that have been deleted from the multimedia learning space intentionally or mistakenly?

These questions are the motivating factors for this research and they were all answered at the end of the project (see section 6.5). The objectives of the project were set so as to ensure that the research questions raised are properly answered. These are discussed in the next section.

1.2 Project Objectives

The goal of this project is to investigate usability in user generated learning spaces. The findings will then be applied in improving the usability of virtual learning spaces in a Personal Learning Environment (PLE). As part of this, work was done on improving the user interface of the multimedia learning spaces in the Manchester PLE; performing development in FLEX 4 and complementary technologies.

In order to ensure the goal of the project is achieved, it was decomposed into broad objectives that acted as guiding posts to achieving the main goal. They are as follows:

- To understand the application of relevant learning theories to the design of learning spaces
- To improve active construction of knowledge by users in virtual learning spaces
- To improve self-directed as well as collaborative learning in virtual learning spaces
- To Investigate (by user evaluation) the usability of current learning spaces in the PLE
- To develop improved user interfaces (that are being tested by users) for the current learning spaces in the PLE
- To assess and improve by user testing, the usability of the developed user interfaces

This project covers a broad spectrum because it involves designing for learning. However, given the time available for the project, a limited scope was defined to ensure its success.

1.3 Project Scope

The project scope includes investigating and improving the usability of virtual learning spaces in the Manchester PLE. Usability evaluation for pedagogical applications comprises two parts namely: technical/functional usability evaluation and pedagogical usability evaluation. This involves the use of testing and formative evaluation in iterative process of design & implementation. A careful comparison of various frameworks for usability testing was made to select the most appropriate one that can be applied. Cooperative evaluation was selected for its inherent benefits. This is further reported in section 6.1. Web usability is also within the scope of the project. This is because virtual learning spaces are mostly implemented on web pages. Consequentially principles of web usability design were incorporated in the design and implementation of the improved learning space. Finally, evaluation of the improved learning interface was carried out periodically using formative evaluation with. Some innovative features of the improved learning space are as follows:

- Personal development planning (PDP) which enables the learner to take control of his learning. This is discussed in details in section 2.5.
- The ability to convert static PDPs to reusable 'Knowledge Artefacts'.
- The ability to track and highlight changes in the space both in private and collaborative mode.
- The transfer of desirable classroom and informal space characteristics into the virtual learning space such as ambience, immersion principle, attention and motivation theories, layout re-configurability, knowledge discovery, etc.

The scope defined above was broadly covered and the target goal of the project was achieved. The rest of this report gives a background of the project as well as design and implementation details. The structure of the report is discussed next.

1.4 Thesis Structure

The project has been introduced; the motivation for it given as well as the major questions that motivated the research. The goal and scope has also been clearly stated. The remaining part of this thesis describes how the goal was achieved and is structured as follows:

Chapter 2 – Background

This section discusses relevant background materials with the aim of situating the project into a wider research theme. Relevant learning theories and how they apply to learning are discussed. Current trends in learning space design are examined and then usability of learning spaces with focus on virtual learning spaces is discussed. Finally, a firm theoretical foundation is laid in support of user generated content in learning spaces as well as personal development plans.

Chapter 3 – System Design

Chapters three and four cover the design phase of the project. Chapter three in particular covers the overall design of the system. Basic assumptions, problem analysis and design goals are discussed first. Architectural analysis, Frameworks, micro-architectures and design pattern (both existing and proposed) involved are also discussed.

Chapter 4 – Components Design

Chapter 4 gives a clear description of design considerations for the main features to be added to the PLE. These include the Twitter Search component, Audio component, Space Painter, and the Personal Development Planner.

Chapter 5 – Implementation

In order to achieve the kind of rich user experience expected of this kind of system, Rich Internet Application (RIA) platforms were chosen. This section sets out with a proper introduction of the tools of the trade; giving justification for the choice of Adobe Flex and complementary technologies. Initial preparatory tasks and the main implementation details are then discussed. The section concludes with the challenges faced during the implementation phase and how they were overcome.

Chapter 6 – Evaluation and Analysis

This section reports the user evaluation of the implemented system. The evaluation method, tasks as well as the results are fully discussed. Recommendations are made for improving the system and a critical analysis is also done on the system.

Chapter 7 – Reflection, Future Work and Conclusion

This section concludes the research project. First of all, a summary of the achievements are listed, followed by recommendations for future work. A reflection on the project as well as the approach is also discussed here. Finally, the thesis is reviewed and concluded.

Appendices

The appendices contain supplementary materials relevant to the project and referenced within the dissertation.

Chapter 2

Background

This chapter explains the background behind this project. Significant learning theories are introduced which generally specify ideal ways to learn and teach. This is followed by discussion on Virtual Learning Spaces. We then turn to the importance of usability in virtual learning spaces and finally conclude with reviewing related work.

2.1 Basic Terminologies

2.1.1 Learning

Being a complex process, it is not easy to define learning. According to Domjan & Burkhard (1993), "Learning is such a common experience that we hardly ever reflect on exactly what we mean when we say that something has been learnt". They went ahead to confirm that "a universally acceptable definition for learning does not exist" (Domjan & Burkhard, 1993). However, in the following definition, they attempted to capture many critical aspects of the concept of learning:

> "Learning is an enduring change in the mechanisms of behaviour involving specific stimuli and/or responses that result from prior experience with those stimuli and responses"

In this definition, Domjan & Burkhard view learning from a behavioural perspective which is usually inadequate in defining learning when considered in isolation. The following definition (commonly used but source cannot be traced yet) provide the missing link by defining learning as "a process that brings together cognitive, emotional, and environmental influences and experiences for acquiring, enhancing, or making changes in one's knowledge, skills, values, and world views". This definition attempts to capture the process as well as the product. A noteworthy fact in the definition is that learning is a product of the interplay between the cognitive, the emotional and the environmental. However, the environment can affect both emotion and cognition, positively or otherwise. It therefore becomes a very important factor in learning.

2.1.2 Cognition

Cognition has to do with “how our brain works or how our mind works” (Leonard, Noh, & Orey, 2007). Cognition is the psychological result of perception, learning and reasoning. To put it in simpler terms, it is the act of knowing. Cognition can also refer to the process of knowing. So in my own words, cognition can be a “process” as well as a “product”. To clarify any ambiguity, when we talk about how the brain works, we are not making a biological reference to the brain, “most cognitive theories are more conceptual and therefore it might be more accurate to talk about how the mind works rather than a biological reference to the brain” (Leonard, Noh, and Orey, 2007).

2.1.3 Theory

The term theory is a frequently used word in everyday vocabulary. However, the meaning of a theory in science is not the same as the colloquial use of the word. Marx (1970) defines a theory as “a provisional explanatory proposition, or set of propositions, concerning some natural phenomena”. Leonard, Noh, and Orey (2007) share this “explanatory” perspective. According to them, a theory is “a hypothesis that describes, speculates, or defines a relationship between a set of facts or phenomena through a body of principles, policies, beliefs, or assumptions”. It follows from both definitions that there exists a subtle tone of assumption in every theory.

2.2 Learning Theories

A Learning theory attempts to help us understand the complex process of learning by describing how people (and animals) learn. Learning theories have two chief values:

- Providing a vocabulary and a conceptual framework for interpreting the examples of learning we observe.
- Suggesting where to look for solutions to practical problems.

Interestingly, the theories do not provide solutions to practical problems. However, it should be noted that they do direct our attention to important variables that are crucial in finding solutions. The meaning of “Learning” has been discussed in a previous section above. Learning theories are generally categorized under three philosophical frameworks namely:

- Behaviourism
- Cognitivism
- Constructivism

2.2.1 Behaviorist Theories

Behaviorism is a learning theory based on the idea that all behaviors are acquired through conditioning which occurs through interaction with the environment. J. B Watson, widely regarded as the father of Behaviorism, defined learning as “a sequence of stimulus and response actions in observable cause and effect relationships” (Chowdhury, 2006). Thus behaviourism assumes that the learner is essentially passive, responding to environmental stimuli. According to LTKB (2011), “the learner starts off as a clean slate (i.e. *tabula rasa*) and behavior is shaped through positive reinforcement or negative reinforcement.” Positive indicates the application of a stimulus while negative indicates withholding of a stimulus, thus learning is observable by the “change in the behavior of the learner” in response to the stimuli (LTKB, 2011) as shown in figure 1 below. There are basically two kinds of conditioning in Behaviorism namely classical conditioning and operant conditioning.

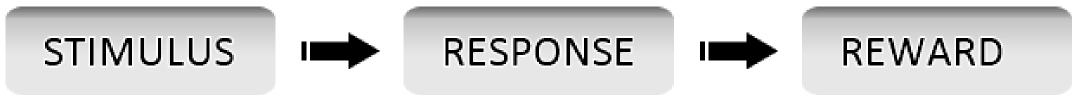


Figure 2.1 - Behaviourist Model

Behaviorism has been applied in the fields of psychology and medicine but our interest lies in its application in learning improvement. Educational approaches such as applied behaviour analysis, curriculum based measurement, and direct instruction have emerged from this model (Kim & Axelrod, 2005). The original theory of behaviorism is now more commonly referred to as “classical behaviorism”. New lines of thought have been extracted from classical behaviorism thus giving rise to Neo-Behaviourism (second Generation) and Social-Behaviorism (Third Generation). Of these two, Social Behaviorism focuses more on learning. It considers learning as a relatively stable behavior modification arising from experience.

2.2.2 Behaviorism in Learning

In Behaviourist approaches, learning is centred on the teacher. The teacher is given the role of transferring his knowledge to the learner which is confirmed done by observing a relative permanent change in the behavior of the learner. This approached is marked by reinforced and programmed learning (LTKB, 2011). According to Standridge (2002), “Behaviorist techniques have long been employed in education to promote behavior that is desirable and discourage that which is not". Below is a summary of some features of a behaviourist learning model:

- Learning is done in small, concrete, progressively sequenced tasks
- Learning is marked by repetition in order to increase retention and speed of learning.
- Consistent use of reinforcements during the teaching-learning process. For instance, with verbal acts such as congratulatory remarks and non verbal reinforcements such as awards.

2.2.3 Cognitivist Theories

Cognitivism as a learning theory looks beyond behaviour to explain "brain" based learning. In other words Cognitivism attempts to improve learning by considering how the human memory works. Cognitivism shares a similarity with behaviourism on the basis that both view knowledge as "given" and "absolute" (LTKB, 2011). However, Cognitivism is based on the assumption that human beings are logical beings and thus make choices that are most sensible to them. Pure cognitive theory largely rejects behaviourism on the basis that behaviorism reduces complex human behavior to simple cause and effect (Fritscher, 2011). However, current trends in past decades have been towards merging the two into a comprehensive "cognitive-behavioural theory" (Fritscher, 2011).

2.2.4 Cognitivism in Learning

Cognitivism approaches learning from a learner-centred perspective. From this perspective, learners need to develop deeper understandings, not just produce the right behaviors (Wortham, 2003). Since these deeper understandings cannot be imposed on learners, they must construct their own mental models with sufficient guide from the teacher. Cognitivism views learning as a change in the learner's understanding, hence the focus is on elaboration. The teacher plays the role of a coach or a facilitator. As a facilitator, he has to provide clues and teach mnemonic strategies (Fortin & Rousseau, 1989), to introduce context. As a coach, he has to constantly evaluate the learner's knowledge to keep the learner as active as possible. Tardif (1992) lists some basic principles that characterize the cognitive learning approach as follows:

- Learning is an active and constructive

- Prior knowledge a crucial factor in learning and believes that knowledge is essentially cumulative.
- Learning permits a link between the new pieces of information and the information already in memory.

2.2.5 Constructivist Theories

Constructivism as a learning theory views knowledge as a “constructed” entity (LTKB, 2011). In contrast to the view that knowledge is absolute and given, constructivism asserts that knowledge is constructed by reflecting on our experiences thus fabricating our own understanding of the world we live in (LTKB, 2011). According to the constructivism paradigm, human learning is an active attempt to construct knowledge based on previous knowledge and the present context. Therefore, every person will construct their own unique set of knowledge. In other words, no two people will start with exactly the same knowledge base, and no two people will construct exactly the same knowledge structures from given experiences or information.

2.2.6 Constructivism in Learning

Constructivism approaches learning from a learner-centred perspective also. However, learning to the constructivist is “discovery and construction of meaning”. In the constructivist view, knowledge cannot be poured in, from one person to another. It holds also, that knowledge does not become part of the learner after memorisation of external objective information but is continuously built as the learner interacts with the outside world thus producing his own interpretations about it. According to DeVries et al. (2002), in most pedagogy based on constructivism, “the teacher's role is not only to observe and assess but to also engage with the students while they are completing activities, wondering aloud and posing questions to the students for promotion of reasoning”. This promotes learning by experimentation and exploration, not by being told what will happen. The constructivist pedagogy involves the following characteristics (Richardson, 2003):

- Student-centredness, evident in attention to the individual and respect for students' backgrounds
- Facilitation of group dialogue that explores an element of the domain with the purpose of leading to the creation and shared understanding of a topic

- Provision of opportunities for students to determine, challenge, change or add to existing beliefs and understandings through engagement in tasks structured for this purpose

2.2.7 Social Constructivism

Social constructivism proceeds from Vygotsky's social development theory. Social development theory argues that social interaction precedes development; consciousness and cognition are the end product of socialization and social behavior (LTKB, 2011a). Vygotsky focused on the connections between people and the socio-cultural context in which they act and interact in shared experiences (Crawford, 1996). It follows from the ideas of Vygotsky and others that learning is a social process. This is why the environment within which learning occurs plays a very important role in social constructivism.

Social constructivism emphasizes the benefits of collaborative learning (Fruchter and Emery, 1999). The role of the educator in this context is to provide what is known as "scaffolding". Scaffolding refers to guidelines and hints which help the learner build strong, complex and relevant ideas (Vygotsky and Cole, 1978). The learner progressively removes this scaffolding and tends towards self-directed learning replacing scaffolding with his own ideas and plans.

2.3 Comparison of Learning Theories

Learning theories are based on different assumptions and focus on different perspectives in explaining learning. Nevertheless, they bear close relationship to one another. Learning styles and behaviours may be viewed as existing on a continuum as shown in figure 2 below. While it may be said that most educational models in use today combine concepts that are mostly drawn from cognitivism and constructivism, that does not mean that the behaviourist theories are not still applicable. For example according to Perraudeau (1996 cited in Ughade et al, 2007), "to develop high intellectual level abilities such as analysis or problem resolution, the teacher will tend to privilege constructivist and cognitivist approaches, whereas for information memorization, a behaviorist approach can be better".

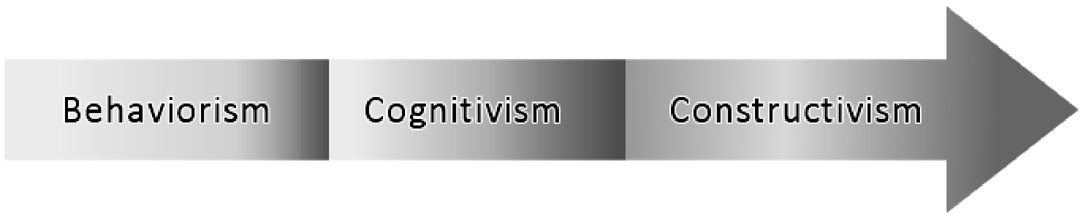


Figure 2.2 - Continuum of Learning Theories

2.4 User Generated Content, Social Learning and Constructionism

The theories of learning discussed thus far attempts to explain how learning takes place. Having understood how it does, we can then improve the learning process in order to improve the product. In this section, the concept of user generated content is discussed and situated within the wider context of social learning environments. Papert's constructionism is then introduced as a way to ensure effectiveness in social learning through user generated content.

2.4.1 User Generated Content and Social Learning Environments

User generated content (UGC) is "one of the main features of this so-called participative web" (Wunsch-Vincent & Vickery, 2007 cited in Clever, Kirchner, Schray, & Schulte, 2009). The term User generated content refers to data, information, or media that is contributed by 'regular people' to the knowledge and information space, usually the web (Krumm, Davies, & Narayanaswami, 2008; Clever, Kirchner, Schray, & Schulte, 2009). The term 'regular people' implies that the contributors need not be experts in the field they are contributing to. They do not need to be expert journalists either; rather they are increasingly being referred to as "the amateur creators" (Wunsch-Vincent & Vickery, 2007). Examples of UGC on the web include ratings and reviews (such as restaurant ratings, Amazon reviews, Google buzz, Rotten Tomatoes, etc), wikis, blogs, forums, videos, online radio, classifieds, shared files, group-based aggregation (e.g. del.icio.us, podcasting), etc.

Although there is no commonly agreed definition of user generated content despite its frequent use (Wunsch-Vincent & Vickery, 2007; Clever, Kirchner, Schray, & Schulte, 2009), the Organisation for Economic Co-operation and Development (OECD) has proposed three main characteristics. These characteristics give a more solid understanding of the concept and they are as follows (Wunsch-Vincent & Vickery, 2007; Clever, Kirchner, Schray, & Schulte, 2009):

1. **Publication Requirement**: UGC has to be published in some context (e.g. on a website or on a social networking site).
2. **Creative Effort**: A "certain amount of creative effort (has to be) put into creating the work" or adapting existing works to construct a new one. This implies that the users must add their own value to the work.
3. **Creation outside of professional routines and practices**: Typically UGC is created without the "expectation of profit or remuneration". The OECD identifies the desire for fame, notoriety or prestige and the desire for self expression as motivating factors in the absence of remuneration.

The third characteristic given by the OECD highlights an important point that provides a common unifying factor for UGC creators and that is 'the desire for self expression'. These users enjoy being creative and entertaining others (Clever, Kirchner, Schray, & Schulte, 2009). They seek the opportunity to share knowledge, experiences and document their lives. These shared values usually bring them together to become part of online communities and collaborative projects. Thus Social Learning Environments (SLEs) are directly or indirectly formed. A Social learning environment provides a space for individuals to collaboratively work and learn together formally or otherwise. Jane Hart, a Social Business Consultant, and Founder of the Centre for Learning & Performance Technologies, suggests a definition for social learning environments that explains almost every facet of the concept. According to her, a social learning environment is:

> "a place where individuals and groups of individuals can come together and co-create content, share knowledge and experiences, and learn from one another to improve their personal and professional productivity; and is also a place that can be used both to extend formal content-based e-learning to provide social interaction with the learners and tutors, as well as to underpin informal learning and working in the organization" (Hart, 2009).

Although Hart's definition talks about the activities in a SLE and the benefits (i.e. learning from each other), it is silent however on the social processes involved as wells as the tools that are required to perform the listed activities. With respect to the required tools, Hart lists 100 tools for learning in another article (Hart, 2011). These include social networking tools, tagging content tools, social bookmarking tools, file sharing tools, blogging tools, etc. The figure below gives a visual representation of the tools necessary for constructing a social learning environment.

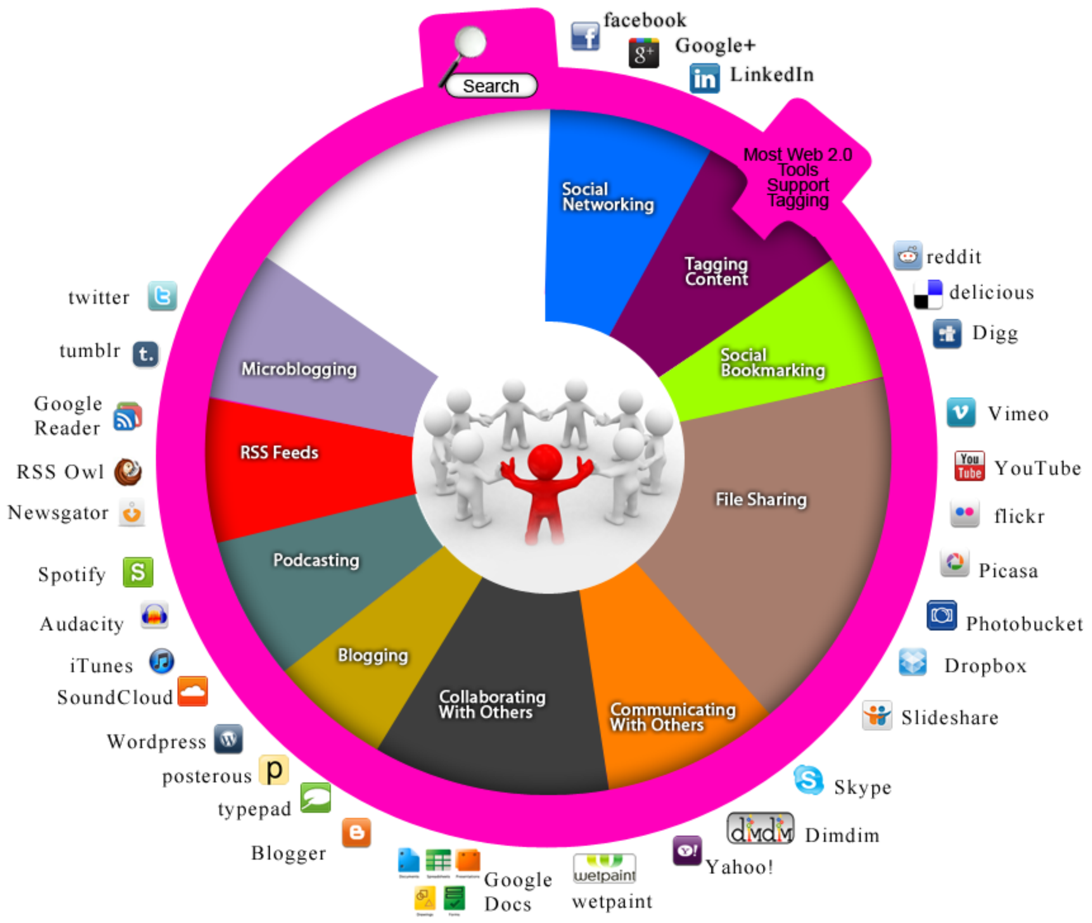


Figure 2.3 - Tools for constructing a social learning environment

(Adapted from Hart's Top 100 tools for learning of 2009 and Kadle's elements for constructing SLEs)

These tools give learners the power to unleash their creativity in various ways. Many learners have used them to create and share different forms of media though sometimes just for the fun of it or to help other learners. However, creating tangible artefacts for the purpose of learning is the focus of Papert's constructionism. The next section discusses how constructionism can be applied in social learning environments.

2.4.2 Constructionism

Constructionism is based on the constructivist learning theory (see section 2.2.5). The constructivist theory suggests that learners construct mental models to understand the world around them. Constructionism takes this further by encouraging the learner to also build a **tangible physical artefact** in the real world to represent the knowledge acquired. Constructionism holds that learning is most effective when part of the learning activity involves the construction of a meaningful product (Papert *et al*., 1986) hence the phrase “learning-by-making” suggested by Papert and Harel (Papert & Harel, 1991). According to them, learning involves "building knowledge structures" and “this happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity, whether it's a sand castle on the beach or a theory of the universe” (Papert & Harel, 1991). In the context of the web, it can be said that constructionism encourages user generated content in the context of learning. In other words, users can generate and publish content for the purpose of learning new ideas or enhancing their skills.

However, Learning is a social process as have already been established in section 2.2.7. Also, in a social learning environment, individuals co-create content in order to share knowledge, experiences, and learn from one another (Hart, 2009). Thus, if we combine Papert’s idea of “learning-by-making” with the idea of “learning-by-sharing” as suggested by social learning environments, we end up with another idea; “learning-by-making-for-sharing”. This idea suggests two possible scenarios:

1. A learner constructs a public artefact for sharing with a community of learners.
2. A community of learners come together to construct a public artefact based on their collective knowledge of a particular subject.

Irrespective of the scenario, a social learning environment can provide the required platform for it to occur. A very useful artefact that can be created to guide the learning process is the personal development plan. Personal development planning deals with how such a plan can be created and used. An interesting idea will be to apply constructionism to personal development planning by converting personal development plans to knowledge artefacts which can be shared and re-used to achieve a learning goal. The next section discusses the concept of personal development planning.

2.5 Personal Development Planning

The Dearing Enquiry (National Committee of Inquiry in Higher Education; 1997) recommended the introduction of ‘progress files’ by universities as a means by which students monitor, build and reflect upon their personal development (Cottrell, 2010; NCIHE, 1997). In 2001, the Quality Assurance Agency for Higher Education (QAA) published the Guidelines for HE Progress Files and now expects all universities to ensure that students undertake personal planning using the progress files (Gosling, 2003; Quinton & Smallbone, 2008; Cottrell, 2010).

Progress files consist of three elements (Cottrell, 2010):

1. personal development processes
2. student records that guide personal reflection and planning
3. the formal university transcript

The processes of Personal Development Planning (PDP) according to Cottrell (2010) are at the heart of the progress files initiative and are the most important aspect of it. Personal Development Planning is defined as

> "A structured and supported process undertaken by an individual to reflect upon their own learning, performance and/or achievement and to plan for their personal, educational and career development" (QAA, 2001; Ward, 2001; THEA, 2011).

Being a structured process implies it has to be organized and where possible integrated into a broader framework. It also implies that specific outcomes be listed for the process. This is probably the reason why the information recorded during the process is usually used to prepare curriculum vitae (CV) for the individual. The definition also talks about the process being supported. According to O'Connell, "early experiences of introducing PADPs at the University of Manchester had shown that students are unwilling to engage in 'standalone' schemes" (O'Connell, 2001). It is therefore important that the processed be structured and supported if the expected outcomes are to be met.

Reflecting upon the learning and making plans for personal, educational and career development is the expected outcome of the process according to the definition. Gosling reveals the underlying activities in the process and also states a way of measuring the success of the process. According to him,

> "PDP refers to a set of activities which engages students in reflecting on their learning and personal goals, <u>creating personal records</u>, <u>planning and monitoring progress</u> towards the achievement of personal objectives and which is intended to improve the capacity of students to articulate their learning goals for themselves and to communicate the outcomes of their learning to others, for example to academic staff and employers" (Gosling, 2003).

Creating personal records is central to the whole process because reflecting and planning is best done using available records of progress. Additionally, progress cannot be effectively monitored in the absence of records of activities. The success of the process according to Gosling can be determined if the students can clearly:

- Articulate their learning goals for themselves
- Communicate the outcomes of their learning to others e.g. academic staff and employers

Communicating the outcomes of their learning to academic staff is usually done by documenting and submitting the progress files to an academic tutor. A brief discussion may follow during which the tutor reviews progress with the student. To the employer on the other hand, a curriculum vita is probably the best way to communicate such outcomes. It can be observed that in both instances, communicating the outcomes of the process involves creating an artefact, the principle of constructionism. This idea is emphasized by Rughani, Franklin, & Dixon (2003) who define PDP as

> "a process by which we identify our educational needs, set ourselves some objectives in relation to these, undertake our educational activities and <u>produce evidence</u> that we have learned something useful" (Rughani, Franklin, & Dixon, 2003).

Their definition also clarifies a fact that is implicit in the other definition, 'identifying the educational needs' of the learner. Several other definitions in the literature of personal learning all agree to the ideas presented so far which can be summarised as follows:

- PDP is a structured process
- PDP involves identifying one's educational needs
- PDP requires reflection upon one's learning, performance and achievements
- PDP requires planning and monitoring progress
- PDP requires some form of recording
- PDP should be supported
- PDP is personal

In the context of social and collaborative learning, the last point can be an obstacle to sharing PDPs. However a closer look at the literature doesn't imply that PDPs are not shareable, rather it implies that subtle modification may be necessary to adapt one learner's PDP to another learner. Sharing PDPs has benefits that cannot be ignored. First of all, it encourages reuse which is important in learning. Also, one learner’s PDP can act as a pointer to the right direction for another learner who is interested in the same subject. Furthermore, shared PDPs can improve the PDP itself since ideas from different learner can be used to modify and improve the PDP. Lastly, if a PDP can be converted to a knowledge artefact, that is, a PDP with a collection of resources for achieving the goals intended, then such knowledge artefact can be preserved, shared and reused as teaching and learning aids in that field. This last point is one of the design goals of this research project. It involves using multimedia components to convert a PDP into a knowledge artefact that can be shared among learners distributed in time and space.

The main purpose of personal development planning is to support learning. The ‘planning’ word is not enough to achieve the expected outcomes, there has to be an ‘executing’ part also. The next section explains how learning is supported using PDPs.

2.5.1 Supporting Learning through PDP

The reflective and planning process in PDP can be framed around the following questions and categories of reflection, review and strategy (Jackson, 2001):

- Where have I been? Retrospective reflection
- Where am I now? Reflection on current situation
- Where do I want to get to? Review of opportunities and identification of personal goals or objectives
- How do I get there? Review of possibilities and decisions on the best way of achieving goals/objectives
- How will I know I've got there? Strategy for setting targets and reviewing progress

These questions can help the learner reflect on what has been learnt prior to the time; what is to be learnt; what is required to facilitate such learning; how to go about the learning process and lastly how to ascertain that the learning goals have been met. The questions raised above by Jackson covers a more generalised approach to supporting learning through PDP. However in an academic context, these questions can be re-framed as follows (Jackson, 2001):

- What have I learnt or done? Retrospective reflection
- What do I need to learn or do to improve myself? Reflection on current situation

- How do I do it? Review of opportunities and identification of personal goals or objectives
- How will I know I've done it? Strategy for setting targets and reviewing progress

Irrespective of the context, the following steps can be identified (Rughani, Franklin, & Dixon, 2003; LLC-UoM, 2010):

- Auditing
- Planning
- Executing
- Reflecting/Evaluating
- Recording

Recording occupies a special place in the listed steps because each step involves some form of recording to be effective. The figure below captures the PDP learning process:

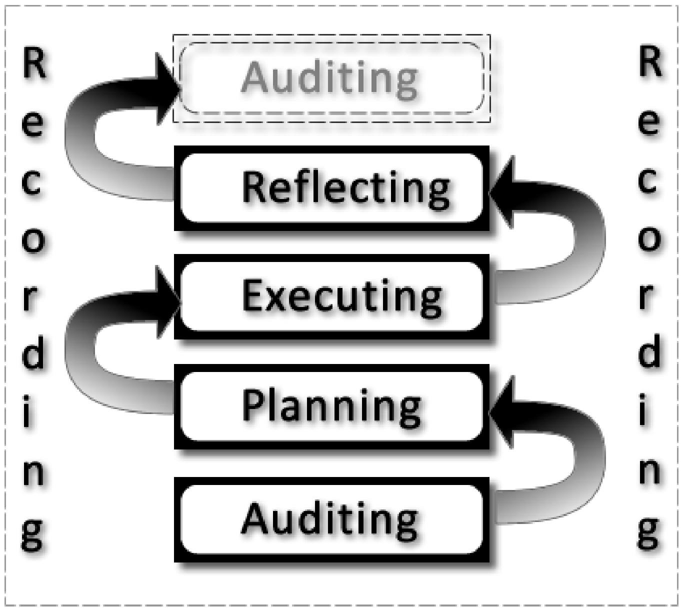


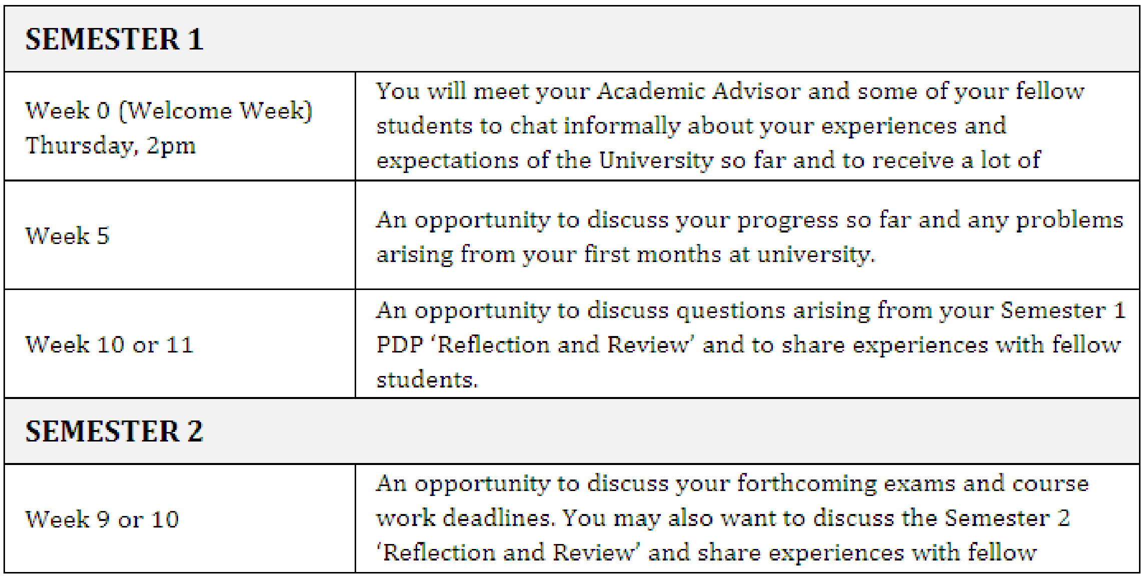
Figure 2.4 - The PDP Learning Process

The figure depicts the fact that the PDP learning process is 'spiral' rather than 'cyclic'. This means that learner begins with auditing and then proceeds to planning, executing, and reflecting on his learning after which the process returns to auditing again. However, we do not view it as a circular process rather it is viewed as a rising spiral indicating the progress the learner is making. In the words of Rughani, Franklin, & Dixon (2003), spiral is used rather than cycle "to emphasise that with every step, we are further on in our educational development than we were at the same time the preceding year".

In an academic environment, a Personal Development Plan is usually a collection of documents to guide the students as well as help them document the process. Each document is designed to be completed at a certain time during the academic year (LLC-UoM, 2010). A good example is the PDP Schedule of the Language and Linguistics centre, University of Manchester (LLC-UoM). The schedule consists of (LLC-UoM, 2010):

- **Skills Audits**: covering topics such as time-management, IT and computer skills, handling information, essay writing, etc.
- **Action Plans**: helping the student to improve these skills
- **Reflection and Review**: recording experiences, and development of academic and transferable skills

At the beginning of each academic year, students are encouraged to download the PDP documents and participate in the process. Academic Advisor meetings are also scheduled for the students where they can discuss their progress, receive guidance and possibly ask questions. A sample schedule for a given semester may look like the table below:



SEMESTER 1	
Week 0 (Welcome Week) Thursday, 2pm	You will meet your Academic Advisor and some of your fellow students to chat informally about your experiences and expectations of the University so far and to receive a lot of
Week 5	An opportunity to discuss your progress so far and any problems arising from your first months at university.
Week 10 or 11	An opportunity to discuss questions arising from your Semester 1 PDP 'Reflection and Review' and to share experiences with fellow students.
SEMESTER 2	
Week 9 or 10	An opportunity to discuss your forthcoming exams and course work deadlines. You may also want to discuss the Semester 2 'Reflection and Review' and share experiences with fellow

Table 2.1 - Sample PDP Schedule

(Adapted from LLC-UoM, 2010)

Unlike the academic context, in a non-academic environment, the learner has to manage this process himself. This can lead to the learning process not being as effective as it should be. However, with support from PDP tools such as the one developed in this

project, the learner can follow the process consistently and, hopefully, without stress. Such tools can also motivate the learner to continue with the process.
The PDP learning process can therefore be summarised in the following steps:

1. Take a skills/knowledge audit
2. Write an action plan
3. Document and keep records as you execute your plans
4. Reflect upon the process, what has been learnt and evaluate the outcome
5. Repeat the steps again in a new direction of learning

PDP is potentially beneficial to the learner, the academic staff and the institution provided the process is carefully implemented and practised.

2.5.2 Potential Benefits of PDP

Personal development planning is beneficial to the student/learner as well as to the institution.

PDP will help students/learners in the following ways (Jackson, 2001; LLC-UoM, 2010):

- integrate their personal and academic development and improve their capacity to plan their own academic programmes
- become an independent learner
- be more effective in monitoring and reviewing their own progress
- be more aware of how they are learning and what different teaching and learning strategies are trying to achieve
- recognise and discuss their own strengths and weaknesses
- develop an increased awareness of their skills
- identify opportunities for learning and personal development outside the curriculum;
- be better prepared for seeking employment or self-employment and be more able to relate what they have learnt to the requirements of employers
- prepare their CV and write good job applications
- be better prepared for the demands of continuing professional or vocational development when they enter employment.

For departments and institutions PDP will (Jackson, 2001; O'Connell, 2001):

- facilitate more effective monitoring of student progress
- result in more effective academic support and guidance systems
- enhance their capacity to demonstrate the quality of support they are giving to students in external review processes.
- Administrative efficiency

Much has been said about 'how' learning is done; however, 'where' learning is done is also an important factor. The environment where learning occurs influences the learning style, process and the outcome. The next section discusses this in details.

2.6 Learning Spaces

In defining the term "learning spaces", Malcolm Brown, started out with a question. "What does the term learning space mean? Why not use a classroom instead?" (Brown, 2005). Learning spaces as defined by Brown (2005) "encompass the full range of places in which learning occurs, from real to virtual; classroom to chat room". According to Brown

> "Just a decade ago, classrooms were the primary locus for learning in higher education. Other spaces included the library, the faculty office (for individual mentoring), and perhaps the café in town. But classrooms were by far the single most important space for learning."

However, a great deal has changed over the years with regards to learning theories, styles and activities. Advancements in learning theories have led to a rethink in designing learning environment. The word "room" (as in classroom, lecture room, etc) is no longer descriptive enough as it has been realised that learning can happen everywhere. The term "Learning Space" is increasingly being used to describe places where learning occurs. Information and Communication Technology has also contributed to changing the notion and location of learning as we shall discuss later, thus leading to the evolution of not only modern physical learning spaces but also virtual learning spaces.

2.6.1 Trends in Learning Space Design

"Learning Spaces often reflect the people and learning approach of the times, so spaces designed in 1956 are not likely to fit perfectly with students in 2006" (Oblinger, 2006a). Consequently, there have been moves to redesign learning spaces not only to conform to the advancements in learning theories but to also conform to the new generation of learners which Oblinger *et al.* choose to call the “Net Generation Learners” (Oblinger & Oblinger, 2005). According to Oblinger (2006a), there are 3 driving forces behind the move to redesign learning spaces viz:

- Changes in students
- Information Technology
- Our understanding of learning

This view is also corroborated by Brown and Long (2006). According to them, “three major trends inform current learning space design” viz:

- Design based on **learning principles** (or theories), resulting in intentional support for social and active learning strategies.
- An emphasis on **human-centered** design
- Increasing ownership of **diverse devices** that may enrich learning.

Their view corroborates Oblinger’s view. These trends, according to them, “have been catalyzed by constructivism, digital technology, and a holistic view of learning”.

The constructivist learning paradigm as earlier discussed focuses on the learner rather than the teacher. Thus in constructivism, we drop the “transmitter-centric” mode of learning in favour of the “active construction of knowledge” by the learner. We drop the focus on “teaching” in favour of the focus on “learning”. This emphasis on learning according to Brown and Long (2006), means that we must also “think about the learner” in designing learning spaces. Learning Spaces, according to them, “are not mere containers for a few, approved activities; instead they provide environments for people”.

Consequently, designing a learning space as an architectural master-piece alone is insufficient for the present day learner. Placing high priority on how the learning space enhances learning is also crucial. This must be what Torin Monahan had in mind when he used the term “**built pedagogy**” to refer to “architectural embodiments of educational philosophies”. In other words, “the ways in which a space is designed shape the learning that happens in that space (Chism, 2006). This holds true regardless of whether the learning space is physical or virtual. Consider the following examples from Chism:

- A room with rows of tablet arm chairs facing an instructor's desk in front of chalkboards conveys the pedagogical approach "I talk or demonstrate; you listen or observe."
- A room of square tables with a chair on each side conveys the importance of teamwork and interaction to learning.

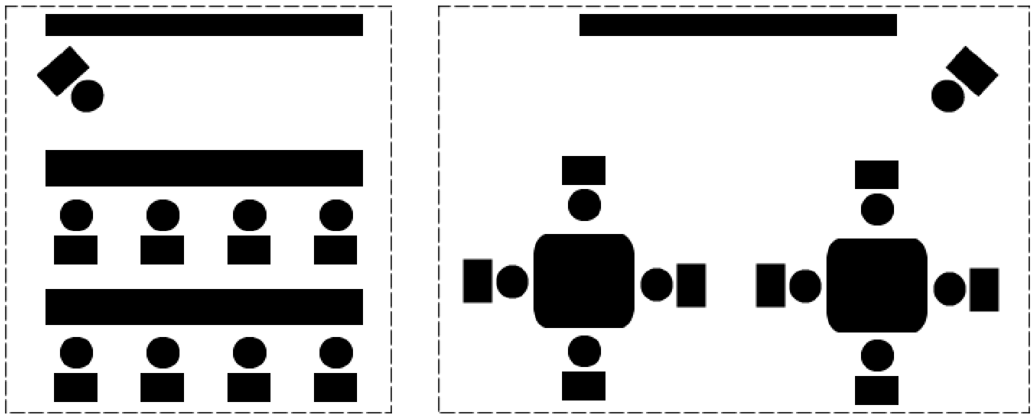


Figure 2.5 - Linear Arrangement vs. Collaborative Arrangement

Present day students do not like the idea of sitting in front of an instructor like dummies and listening "attentively" to the teaching. Their attention shifts quickly from the instructor to other items such as their mobile devices, course mates, etc. Such learners will definitely not fancy the first example given by Chism above. Oblinger (2006a) describes this kind of learners as favouring "active, participatory and experiential learning". This kind of learning according to Neill & Etheridge (2008), "requires a flexible space", and as such the second example will appeal to them since it is more natural to the learning styles they exhibit in their personal lives. In terms of virtual spaces, the first example given in Figure 2.5 above can be compared to software that focus on passing knowledge to the learner while he sits and listens or watches the screen. The learner soon becomes bored and abandons the process. The second example can be compared to social learning environments where the learner collaborates with other learners to discover and construct knowledge. The latter is preferred choice for the present day learners.

Information Technology is also another very potent factor shaping the trends in learning space designs. Trends in Information and Communication Technology continuously redefine the meaning, boundaries and styles of learning. The recent proliferation of low-cost devices as well as the integration of platforms has given learners a whole new universe of learning – learning that is distributed in "time" and "space". We focus on this in the next section when we discuss virtual learning spaces. One big problem with technology is the pace of change. The unrelenting pace of technology change, according to Brown and Long (2006) "can make IT decisions rapidly obsolete". Interestingly, "While platforms and applications come and go, the psychology of how people learn does not" (Brown & Long, 2006). This is why the field of 'Instructional Technology' focuses on adapting the changing technology to fit the psychology of learning. According to Jonassen & Land (2000), "Technology foundations determine what is technologically possible, but grounded practice requires determination of how capabilities should be exploited". Rather than designing learning to suit the technology, the trends with regards to technology focuses on:

- Designing technology to support Learning
- Adapting technology (new and existing) to encourage active, collaborative and experiential learning such as the use of web 2.0 tools, podcasting, mobile devices, social networks, etc.

Current trends in learning space design show the following desired characteristics (Chism, 2006):

- Flexibility
- Comfort
- Sensory Simulation – Colours, Lighting, Ambience
- Technology Support and
- Decentredness

A flexible learning space "better enables innovative approaches to teaching and learning when compared to the traditional classroom" (Neill & Etheridge, 2008). With the right approach, "the entire campus can become a learning space" (Mitchell, 2004). Indeed, in the virtual approach adopted here, both on and off campus learning experiences are to be supported. The 3 trends discussed here underlie this emerging reality (Brown & Long, 2006) while the desired characteristics stated above are the features these trends will produce to support active, participatory and experiential learning. The next sections concentrate more on virtual learning spaces which is the focal point of this project.

2.6.2 Virtual Learning Spaces

While physical spaces are tangible and fixed in time and space, virtual learning spaces[2] are the direct opposites. Also, whereas physical spaces exist around us, virtual spaces exist on machines and devices. While there is no single definition for a virtual learning space, most writers define them by specifying their purpose, components and characteristics. van Harmelen (2011) defines it as "a place where one or more learners can assemble learning materials that are relevant to the pursuit of their learning goals". Such learning materials might be quite diverse. For example, they could be something like a personal development plan which expiates both learning goals and learning strategies. Alternatively, the content might be a corpus of student work, generated on a particular topic (van Harmelen, 2011), or a mixture of both.

It follows therefore that virtual learning spaces are "**designed**" to support and enhance learning through the use of computers, multi-media devices, mobile devices and other technology based tools. Dillenbourg, Schneider & Synteta (2002) suggest the following about virtual learning spaces:

- A virtual learning space is a designed information space.
- It is a social space: educational interactions occur in the environment, turning spaces into places.
- The virtual space is explicitly represented: the representation of this information/social space can vary from text to 3D immersive worlds.
- Students are not only active, but also actors: they co-construct the virtual space.
- Virtual learning spaces integrate heterogeneous technologies and multiple pedagogical approaches.
- Virtual learning spaces overlap with physical environments.

Virtual Spaces are continually being improved to support active, collaborative and experiential learning. The goal in improving virtual spaces is not to use them as replacements for physical spaces as some might wrongly envisage, rather both spaces are meant to be complementary.

[2] To eliminate all doubts, the term "Virtual Learning Space" does not in any way suggest that the "learning" is virtual and not original or authentic. It is the "environment" that is virtual, not the learning. I personally will like to call it "Virtual Spaces for Learning" but for purposes of consistency with the literature stick with "Virtual Learning Spaces".

2.6.3 Usability of Learning Spaces

Usability is defined as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use" (ISO 9241-151, 2008). Usability is a "Quality" attribute, therefore in order to define clearly what usability implies in any context, some form of concrete criteria or attributes must be defined through which usability can be measured. A popular model for doing this is based on five quality components or criteria namely learnability, efficiency, memorability, errors and satisfaction.

Jakob Nielsen is a renowned authority in usability engineering. He observes that usability is a "narrow concern compared to the larger issue of system acceptability" (Nielsen, 1993). The diagram below shows the position of usability with regards to system acceptability.

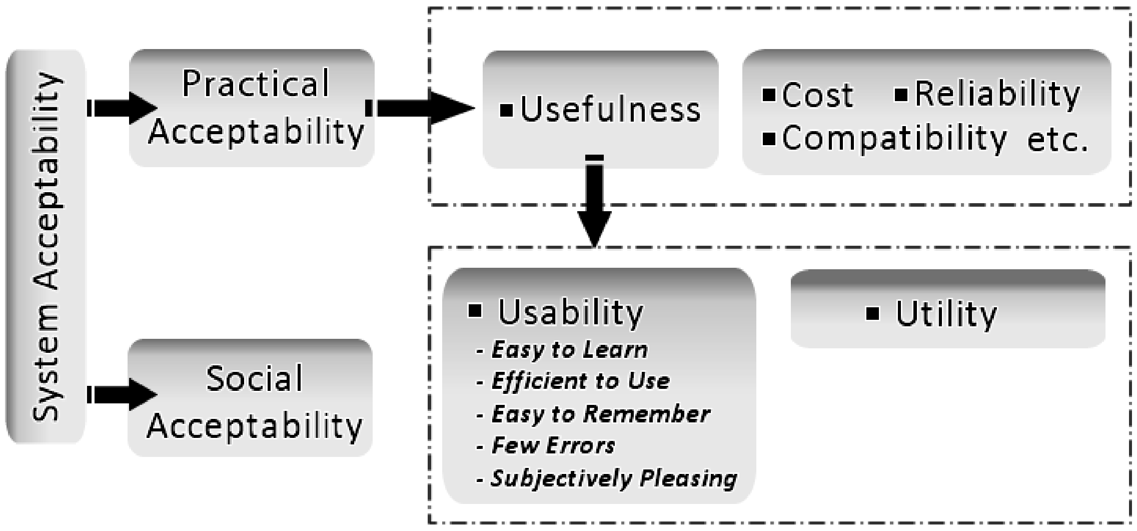


Figure 2.6 - Attributes of System Acceptability

System acceptability according to Nielsen (1993) "is the question of whether the system is good enough to satisfy all the needs and requirements of the users and other potential stakeholders." Generally, a system that will be able to "satisfy all" will be a utopian dream. Usually, there is an acceptable level of satisfaction that a system is required to meet. In the framework of social acceptability proposed by Nielsen and corroborated by Ben Shneiderman (1980), Usability, is a defining component of "Usefulness" and is composed of the five attributes identified above which are described below:

- LEARNABILITY: How easy it is for the user to learn to use the system. According to Nielsen (1993), “The system should be easy to learn so that the user can rapidly start getting some work done with it”.
- EFFICIENCY: The level of productivity in use after learning to use the system. In other words, how quickly can the user perform tasks? A high level of productivity is desired in this case.
- MEMORABILITY: The system according to Nielsen, “should be easy to remember”. This will enable the user return to the system after a period of not using it and re-establish proficiency without having to learn about the system from first principles again.
- ERRORS: The error rate of the system should be very low. This does not imply that errors may not occur but if and when they do, how severe are they and how easy it is for the user to recover “gracefully” from these errors. For Nielsen, “catastrophic errors” must not occur.
- SATISFACTION: This is a measure of how “pleasant” it is to use the system. Among all the criteria, this is the most subjective one and is not easy to measure.

For learning support systems, Nokelainen (2006) expanded Nielsen’s usability model by adding “Pedagogical Usability” to the “Utility” branch of the System Acceptability tree and renaming usability to “technical usability”. Nokelainen defines Pedagogical Usability as being “dependent on the goals set for a learning situation by the student and teacher”. It follows thus that evaluating the usability of a virtual learning space becomes more challenging since the technical usability alone is not enough. The environment must also meet the pedagogical demands in terms of achieving learning goals. How do we measure and ascertain that these learning goals have been achieved? Zaharias & Poylymenakou (2009) agree that “evaluating the usability of e-learning applications is not a trivial task”. In order to do a successful usability evaluation, the users, task and context must be identified. According to Zaharias and Poylymenakou, in terms of e-learning, “the main task for the user is to learn, which is rather tacit and abstract in nature”. I believe this is why most e-learning tools have poor usability records. They are either not being evaluated for usability at all or the evaluation is not properly done. To develop an effective usability evaluation framework for virtual learning spaces, the evaluator must familiarize himself with learning theories, learning styles and educational testing research (Zaharias & Poylymenakou, 2009). Three widely used methods for usability evaluation (Hertzum & Jacobsen, 2001) are

- Think Aloud (TA)
- Heuristic Evaluation (HE)
- Cognitive Walkthrough (CW).

Other methods are Questionnaires, Direct Observation, Interviews, and Focus Groups.

2.7 Existing PLE Architecture to Support Virtual Learning Spaces

The Manchester Personal Learning Environment (MPLE) is an integrated PLE that aims at providing machine support for people to learn together in collocated and distributed settings (van Harmelen, 2010). Its design is based on social constructivism; Papert's constructionism and self-directed learning (see sections 2.2.7 and 2.4.2). The MPLE provides a social networking service layer; tailored for educational purposes. It also contains multi-user, multi-media spaces that can either be used for personal learning or learning in groups. PLE users may either create and use their own spaces, or join a group with other learners, and meet in community created spaces to pursue common learning activities and realise learning goals (van Harmelen, 2009).

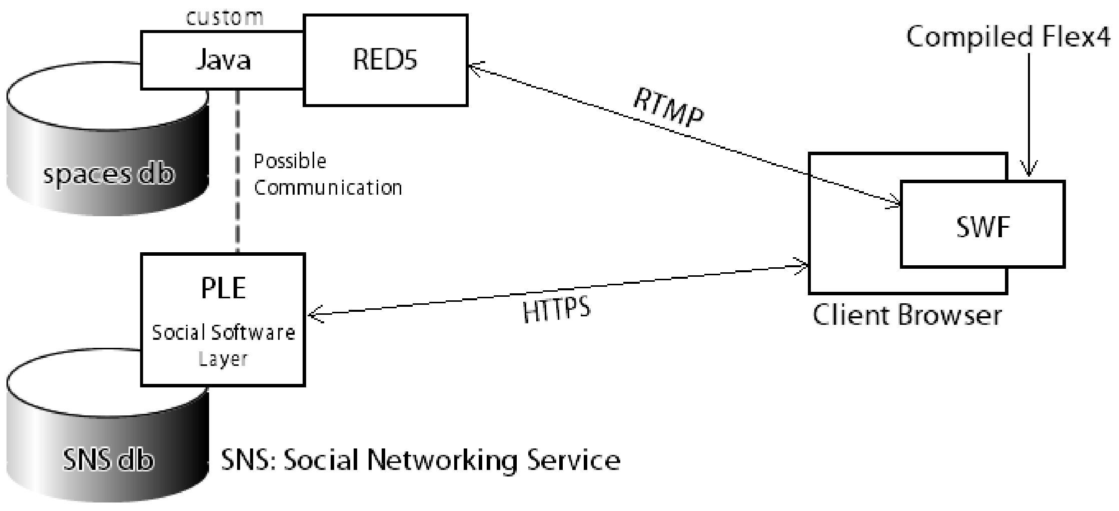


Figure 2.7 - Overall Architecture of MPLE

The diagram above depicts the architecture of the current MPLE in relation to the media spaces. The media space is written in Flex which compiles to SWF format. This format is executable in Adobe Flash runtime environment available in major web browsers. The spaces connect to a Red5 back end which uses Real Time Messaging Protocol (RMTP) for streaming multi-media content to the spaces. Red5 is written in Java and supports live stream publishing. This enables concurrent editing and updating within the spaces. In previous versions of the Manchester PLE there was some communication from the spaces backend to the PLE social software layer, but this is not exploited currently. Such communication does not involve real-time streaming of media space content. An example can be a notification feed to a user about changes in a shared space.

This architecture will be preserved, as will the use of the Red5 backend. The latter would have required modification if this work extended to re-establishing feeds from the spaces to the social software layer, but time precluded this.

2.8 Summary

This chapter has presented the background research carried out to understand the project in the wider context of psychology and pedagogy. The basic terminologies relevant for understanding the discussion were defined. Learning theories and their application in learning were explained. The concepts of user generated content; social learning environments and constructionism were also explained. Personal development planning (PDP) which enables the learner to take active control of his learning was explored; showing how it can support learning and the potential benefits it brings to the learning process. Since learning spaces (whether physical or virtual) are where learning happens, the chapter also extensively discussed learning spaces. This discussion begins with trends informing the design of learning spaces which include changes in students, technology and our understanding of learning; then proceeds to explain virtual learning spaces and how usable they are. The concept of usability is examined within the wider context of system acceptability. Finally, the existing PLE architecture to support virtual learning spaces is fully described.

The design and implementation done in this project are based on the concepts discussed in this chapter. Great spaces foster great learning whether on physical or virtual spaces. Learning spaces should therefore be designed to enhance learning. Doing this requires an understanding of the principles behind learning which are embodied in learning theories, styles and processes.

Chapter 3

System Design

This section covers the overall design of the system. Due to the nature of work to be done, the design phase was divided into two parts. The first part which this section reports, covers the initial considerations; problem analysis, and design goals. It also covers the layout and architecture of the system as a whole. Subsequent sections describe the design of individual components within the system.

3.1 Initial Considerations and Assumptions

As already discussed in section 1.2, the main focus of the project is improving usability in user generated learning spaces. User generated learning spaces according to van Harmelen (2011) are "(learning) spaces which are populated with content by one or more learners". This brings some concepts as well as questions to mind but before we attempt to outline any of these, it is very important to clearly state some implicit facts with regards to this definition. The basic assumptions are as follows:

i. The content in the learning spaces can come from a variety of sources
ii. The content in the learning spaces needs to be persisted so the learner can continue adding to the knowledge base rather than starting from first principles each time
iii. The space is meant to be a collaborative space. This implies that the learner needs to be able to share the content(s) of the space with other learners.
iv. In a collaborative mode, other learners should be able to contribute to the content by either generating additional content themselves or modifying already existing content.
v. In collaborative mode, there should be a way to synchronise activities between the learners to ensure that one learner's activity does not over-write the others'

In order to ensure that the goals of the project are met, it is important to design the system such that it is easily extensible in the future and this requires a careful analysis of the challenges involved in designing this kind of interactive, collaborative system.

3.2 Problem Analysis

There is no doubt about the fact that learning with multimedia is fun, however, according to Lindgaard, Brown, & Bronsther (2005), “it takes more than an understanding of technology to design useful and usable multimedia interfaces”. The additional requirement Lindgaard *et al.* were referring to is a broad understanding of useful learning theories as covered in previous sections. Additionally, they also remarked that one must understand “cognitive models describing how sensory information may be perceived, interpreted, stored, and retrieved when we need it” (Lindgaard, Brown, & Bronsther, 2005). The role of animated and static graphics, the amount of information that should be presented on a page, the ease of use of the system, the visual cues and interactive responses etc all impact the cognitive state of the learner while using the multimedia learning space. All these should be considered carefully in the analysis to ensure the end product is not only a product of technical competence but also pedagogically useful and usable.

Research into the learning literature shows that Learning may be described as a three-phased process (Mayer 2002 cited in Lindgaard, Brown, & Bronsther, 2005):

i. The learner should select relevant material(s);
ii. The learner should organise the selected material; and
iii. The learner should integrate it with existing knowledge.

The first phase of this process is the responsibility of the learner. It can sometimes be very easy for him to select relevant materials to support his learning; however, in situations where he cannot easily find relevant materials, a search using the internet, library resources, asking other learners, etc can lead to discovery of relevant materials. The second phase is usually where user actions become sometimes tricky and complicated. The materials selected or discovered need to be organized is such a way that the following can be met:

i. The materials can be easily identified
ii. The relevance of the materials can be easily pointed out
iii. The relationship among the materials identified can be easily understood
iv. The sources of the materials can be readily identified for further research.

The third phase is more of a cognitive process, integration of new knowedge with existing knowledge. Given that the second phase is well handled, the third should happen smoothly and effectively. However, learning processes can and do go wrong, so, I propose a fourth reflection phase which should provide a check on the third phase having been achieved. If little or no integration of new knowledge was achieved, the fourth phase when done will result in information to allow the process to be repeated more effectively. This phase is known as the 'reflection' phase and it involves the learner objectively reflecting on the goals that motivated the learning process in order to ensure that they were achieved. The figure below captures the phases. It shows that reflection should not be left till the end, rather every phase can benefit from some form of reflection to improve that phase.

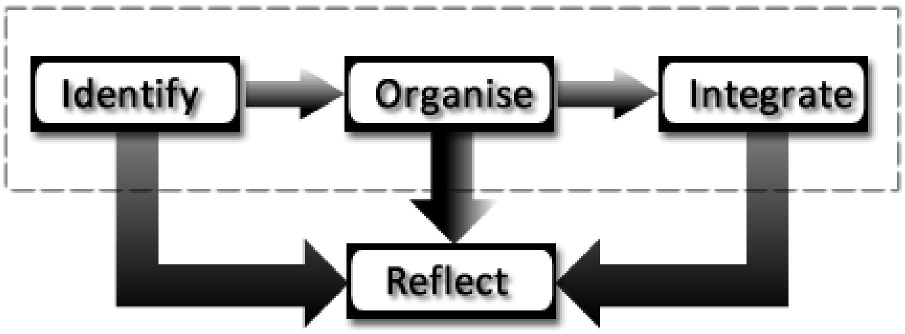


Figure 3.1 - Learning Phases

In order to enable the learner organise his learning materials and easily integrate them into existing knowledge, the multimedia space had to be equipped with basic building blocks that the learner can use to achieve these. These basic building blocks are the media components, including edges, and the toolbox.

3.2.1 The Basic Building Blocks

The media components are laid out on a drawing canvas (referred to here as a **'Whiteboard'**) and organized by dragging, dropping and interconnecting them using directional and non-directional edges. The relationship between the Whiteboard, media components and edges is shown in the figure below:

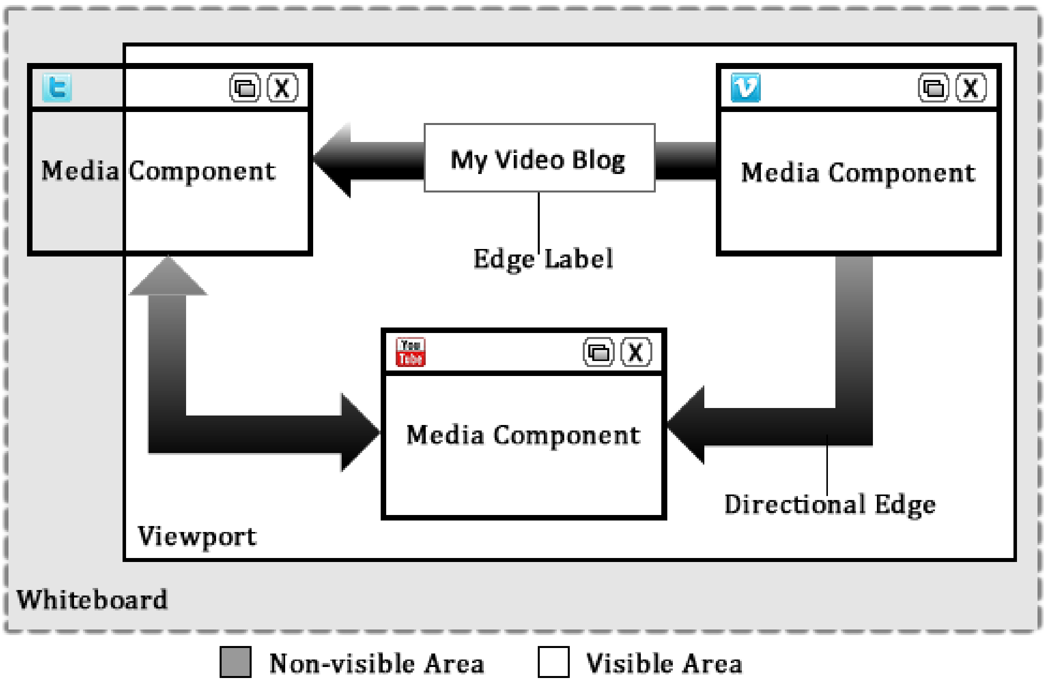


Figure 3.2 - Whiteboard, Components and Edges

The diagram shows that the whiteboard can be an unlimited span or area where components can be laid out and interconnected to form an organised body of knowledge. Notice however that the entire whiteboard need not be seen all the time; a viewport is used to visualize a portion of the whiteboard at a time. To clarify any doubt, a viewport is a framed area on a display screen for viewing a specified portion of the whiteboard at any given time. The part of the screen that is visible is the part covered by the area of the viewport. Other parts are invisible until they are brought into view either as a result of a user action or automatically in response to some system event. The user can bring other parts of the whiteboard into focus in the viewport by scrolling the viewport in a horizontal or vertical manner. A better way to do this will be to enable dragging parts of the whiteboard into view. This will eliminate the scrollbars and also provide a more interactive user-friendly experience to the user. This is because while scrollbars need to be activated at specific points to trigger a response, dragging can be done anywhere on the whiteboard that is unoccupied by a component or an edge.

As shown in Figure 3.2 above, the components are displayed on the whiteboard and where necessary, are interconnected using edge lines. An edge line can have a single directional arrow head, bi-directional arrow heads or no arrow head at all. This gives the learner the ability to represent the relationship between the components clearly and further helps in the organization phase of learning.

The toolbox is also one of the basic building blocks. It is a collection of icons which are pictorial representations of the media components. The user elects to display a media component by activating its icon in the toolbox. Figure 3.3 below shows a sketch of the toolbox as well as a screenshot of its implementation.

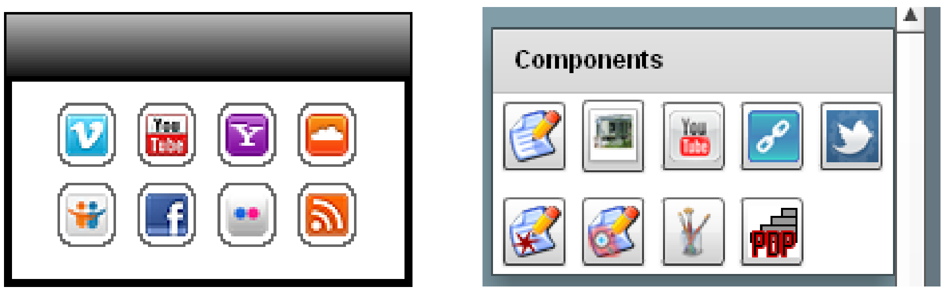


Figure 3.3 - Sketch and Actual Implementation of the Toolbox

The proposed components are as follows (ones implemented in this work are marked with a *):

- Auto Save textbox component*
- Fully expandable component*
- Twitter search component*
- Audio component*
- Space-painter component*
- Personal Development Planning (PDP) component*
- Image component
- YouTube video component
- Links component

Having analysed the problem; devised the basic building blocks which can be used in solving it and identifying the proposed components, the next section examines the design goals of the system.

3.3 Design Goals

The design goals of the system revolve around the main objectives of the project as listed in section 1.2. It is important to have clear design goals so as to be able to focus the design and implementation of the system in line with the expected outcome.

The main design goal of the multimedia learning space is to allow learners represent knowledge in various formats using visual elements. These visual elements can be textual or graphical. They can be static or animated. In order to achieve this, the elements are represented as components on a whiteboard as described in the problem analysis section (section 3.2). The design of the components should be such that the user can easily drop a component on any part of the whiteboard or drag to reposition them on a desired area of the whiteboard. Also, the component should be able to exist in different sizes so that the user can minimize, maximize or drag an edge of the component to resize it as desired. Finally, with regards to the components, it should be easy to remove a component from the whiteboard when it is no longer needed by the user.

One thing that is yet to be mentioned so far is the process for adding a component to the whiteboard. In the existing implementation, the user is clicks an iconic representation of the component in the 'toolbox'. The specified component is then added in the middle of a viewport. This implementation was maintained for this project. Another way to add a component might be to select its iconic representation; then drag to lay out the component to a given desired dimension on the whiteboard.

In terms of relationships, edges should be drawn to link components together. The design goals with respect to this are as follows:

i. It should be possible to link one component to many other components
ii. It should be possible to define a direction for the relationship using arrow heads which can either be unidirectional or bidirectional.
iii. It should be possible to label edges using free text to annotate the relationship between components.
iv. It should be possible to route edges so that they can trace a flexible path between two components.

There are also design goals for the system in terms of interaction and collaboration.

One such design goal is the ability to use a remote cursor controller to track the movement of other users' mice around the whiteboard. This is a useful feature as each user will be able to see the position of other users' mouse cursors and as such be able to pre-empt their actions. Additionally, during a presentation, a user can use such functionality to point to elements on the screen while explaining them to the remote audience. The remote cursor can be displayed as a shape (e.g. a small oval or rectangular shape) on the whiteboard but an additional design goal is to attach the user identity to it such that in the presence of multiple remote cursors, the user can distinguish one from another.

Another design goal with respect to user interaction and collaboration is the ability to synchronise activities and changes in the learning space among collaborative users. Changes initiated by one user should be immediately updated on the screens of other users. This task requires a careful design of the system as it can be tricky to do this due to synchronization concerns that include locking mechanisms and network considerations.

The design goals discussed so far might not have been easily achieved given the time frame for the project. Fortunately, some have already been realised in the existing system. Extending and improving the existing system in order to achieve the other design goals therefore became a less difficult task. The next section examines the architecture of the existing system to give a clearer picture of how some of these goals were achieved.

3.4 Architecture and Layout

The architecture of the existing system was carefully designed in line with the goals discussed above. The design ensured that every part of the system is cohesive and well decoupled from other parts to foster extensibility. This section gives a clear description of how the system is structured and describes each part that makes up the system. Frameworks, design patterns and concepts used are fully explored.

3.4.1 Media Spaces Architectural Overview

In section 2.7, the overall architecture of the Manchester Personal Learning Environment (MPLE) was discussed, describing clearly the relationship between the multimedia learning space and other parts of the PLE. In this section, the architecture of the multimedia learning space is discussed showing the different components that work together to deliver the functionality of the application.

The learning space is made up of three components namely:

1. The Media Server
2. The Space Client
3. The RTMP Connector

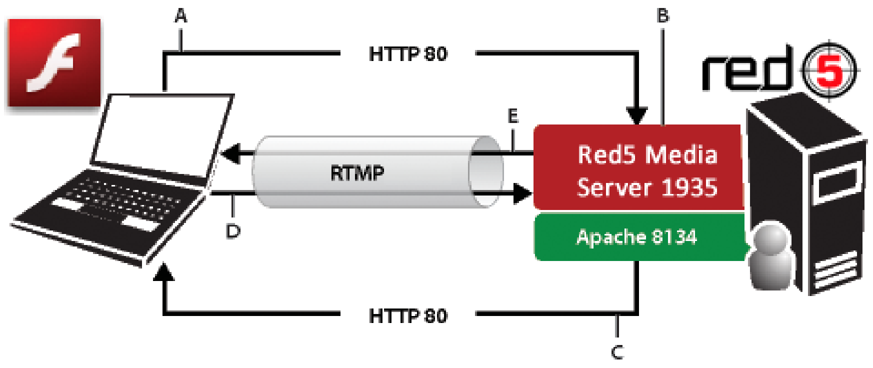


Figure 3.4 - Media Space Architecture

(Adapted from Adobe Core Server Architecture; Adobe, 2009c)

The figure above depicts the interaction between the 3 components. Each labelled interaction is explained as follows (Adobe, 2009):

- **A.** The client application (compiled SWF file) executing in Adobe Flash run-time makes a request to the server for content over HTTP. Authentication can be done if necessary.
- **B.** The request is passed by proxy to Apache Web Server.
- **C.** The web server (Apache) delivers the requested content to the client
- **D.** The client application makes a request for content over Real Time Messaging Protocol (RTMP)
- **E.** The Red5 media server progressively streams content to client application

The media server is a connection hub. The basic principle behind the media server is that it acts as a central resource point for all connecting clients and thus enables them to share real time information as well as enjoy integrated real-time communication. Using a media server, an application can stream live and on-demand content quickly and easily to a wide variety of platforms and devices (Adobe, 2011b). The following are some of the things that can be done using a media server:

- Live Stream Publishing
- Streaming Video (FLV, F4V, MP4, 3GP) e.g. Live video broadcasts
- Streaming Audio (MP3, F4A, M4A, AAC)
- Recording Client Streams
- Shared Objects which allows clients to interactively share resources in real time
- Remoting which provides methods to send data back and forth from Flash applications to the server

Flash-based applications connect to the media server using Real Time Messaging Protocol (RTMP). This is discussed further below. When connected, the server can exchange (send and receive) data with the connected client applications. It can also invoke methods on target clients. The clients on the other hand can initiate Remote Procedure Calls (RPC) on the server side (Adobe, 2011). Data is transported between clients and server using standard ActionScript objects in Action Message Format (AMF) or other supported formats. Examples of media servers include:

- Adobe Flash Media Server (http://www.adobe.com/products/flashmediaserver/)
- Red5 Open Source Media Server (http://trac.red5.org/)
- Wowza Media Server (http://www.wowza.com/)

The existing system uses the Red5 open source media server.

The browser-based space client is written in ActionScript 3.0 and Adobe Flex MXML which is compiled to a Flash SWF that is executed in the browser. This client is the interface between the user and the system. Components are instantiated and organised on the client application while data needed to populate these components are persisted on the server in a database. Due to the collaborative nature of the system, data must be persisted in the server immediately they are inputted into the client components. This will allow other users of the space to be notified of the changes and be updated in real time. A screen shot of the existing space client is shown below.

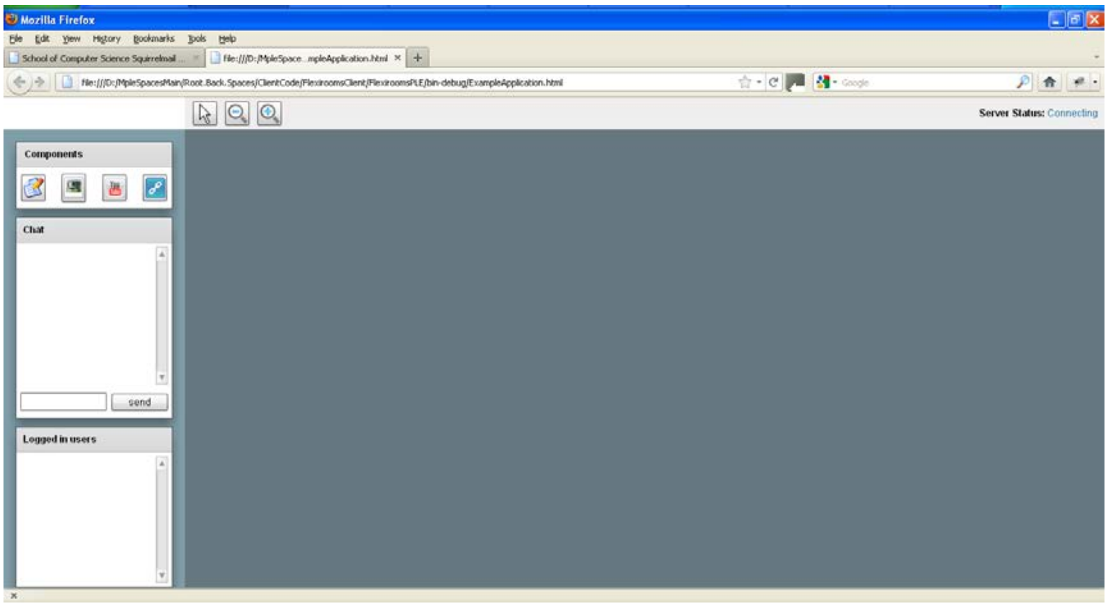

Mozilla Firefox
File Edit View History Bookmarks Tools Help
School of Computer Science Squirrelmail ...
file:///D:/MpleSpace...mpleApplication.html
file:///D:/MpleSpacesMain/Root.Back.Spaces/ClientCode/FlexroomsClient/FlexroomsPLE/bin-debug/ExampleApplication.html
Google
Server Status: Connecting
Components
Chat
send
Logged in users

Figure 3.5 - A screen shot of the existing space client application

The Real Time Messaging Protocol (RTMP) connector also referred to as the 'Red5Connector' in this system, manages the communication between the media server and the client application. RTMP is a protocol initially developed by Macromedia (acquired by Adobe) for "high-performance transmission of audio, video, and data between Adobe Flash Platform technologies, including Adobe Flash Player and Adobe AIR" (Adobe, 2011c). The specification of the protocol was later released by Adobe for public use.

CONNECTION	DESCRIPTION	MEDIA SERVERS
RTMP	Real Time Messaging Protocol	FMS, Red5, Wowza
RTMPT	RTMP (Tunnelled over HTTP)	FMS, Red5, Wowza
RTMPS	RTMP over SSL or RTMP (Secure)	FMS, Red5, Wowza
RTMPE / RTMPTE	128-bit encrypted RTMP (Encrypted and Enhanced); RTMPTE (Tunnelled over HTTP)	FMS, Red5, Wowza
RTMFP	Real Time Media Flow Protocol	FMS

Table 3.1 - RTMP Connections and Supporting Media Servers

(Data source – AskMeFlash (AskMeFlash, 2009); Adobe (Adobe 2009, 2011b, 2011c), Red5 (Red5, 2010))

Table 3.1 above lists different types of RTMP connections and the media servers supporting them. The connection type used in the existing system is RTMP.

The architectural structure of the multimedia learning space is very flexible such that any of the components can be easily extended or even replaced to give the system enhanced functionalities. A good example can be seen in this project which focuses development on the space client. This includes improving the existing components, adding new components and also enhancing the user interaction experience of the space client. The subsequent sections take a closer look at this part of the architecture, layout, structure and design details.

3.4.2 The Space Client Layout

The visual layout of the space client is shown below in Figure 3.6.

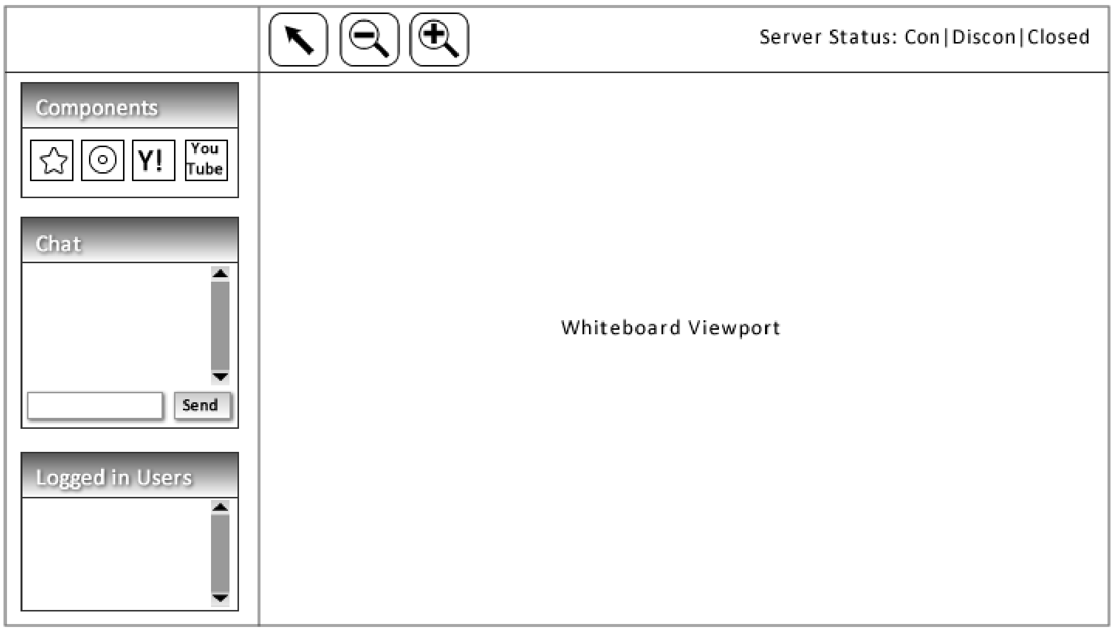


Figure 3.6 - Space Client Layout

The space client is made up of three sections namely:

1. The Left Sidebar
2. The Top bar
3. The Whiteboard

The left side bar houses the components toolbox, the chats widget and the connected users' listbox. Icons on the toolbox represent various components and when activated by the user (e.g. via a mouse click), the corresponding component is instantiated on the whiteboard.

The top bar houses the remote cursor control and the zoom tools. The remote cursor control is used by the current user to signal to the system that his mouse cursor position should be tracked and displayed to other connected users. This control is a toggle control which means that the first time it is clicked, it turns on the functionality. The next time, it turns it off and so on. The zoom controls on the other hand can be used to increase or decrease the scale factor of the whiteboard viewport thereby making the components appear bigger or smaller on the screen. The extreme right side of the top bar also shows the connection status of the application to the server. The connection status can either be 'Connecting', 'Connection failed', 'Connected', or 'Connection Closed'.

The whiteboard is the main location where components are displayed, interconnected and organised. As previously described, the whiteboard has a much wider size than the available screen size and as such is being viewed through a rectangular frame referred to as a viewport. Hidden portions of the whiteboard can be brought into view by dragging. This is a mouse operation that involves pressing down the mouse button and moving the mouse while holding the button down.

The layout of the interface is achieved by using flex layouts and containers. The three components are flex canvas components and they are laid out using flex group containers. The positioning of the components is carefully done using the system screen coordinates with the origin position (x,y = 0,0) at the upper left corner of the screen. In designing a visual multimedia application, two things are very important; one of them is a simple, easy to use layout with graphically appealing visual components. The other is a carefully designed implementation of the functionalities of these components so as to make it easy for the user to achieve his desired goals. In this case, a carefully designed architecture (the MVCS architecture) is employed which is described next.

3.4.3 The MVCS Architecture

An architectural pattern helps in decomposing a complex system into simpler ones with “Responsibilities, Relationships and Interactions” (Berkovitz, 2006). An example is the Model, View, Controller, Service (MVCS) architectural pattern. It is employed to achieve a clear separation of responsibilities in the design of the space client. When a system such as this is being designed, it is important that “cohesion” is increased while “coupling” among components is reduced as much as possible. This promotes reusability because the interdependency among components is greatly reduced.

The figure below shows a typical structure of the MVCS architecture. Each of the component parts of the architecture is described below.

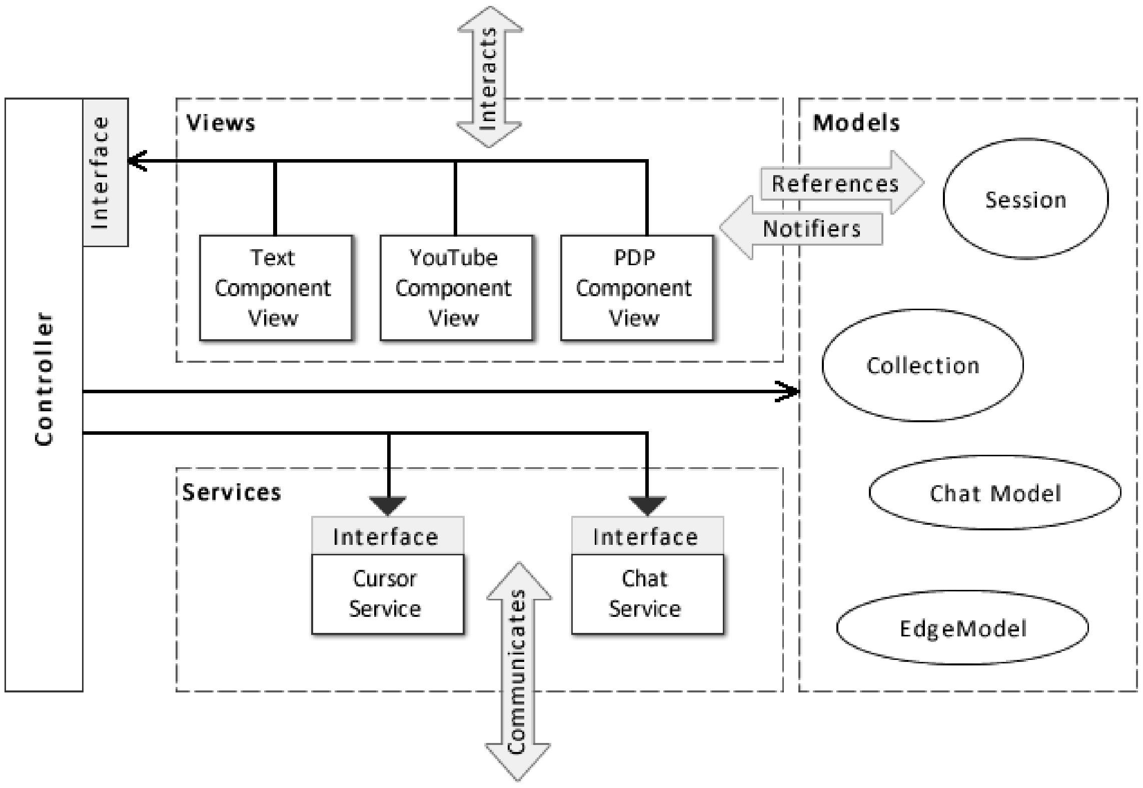


Figure 3.7 - The MVCS Architecture

The **Model** is the data store. Its function is to capture and store the state of the application. The state can be represented as objects, collections, properties, etc. The model also watches the stored data and any change to the state causes it to dispatch events notifying the **View** of such changes. The model does not reference any non-model application component (Berkovitz, 2006).

The **View** handles the presentation and interaction. Presenting the state of the application to the user in its raw form (as objects, arrays, etc) will make no meaning to the user so the view presents it in a way that the user can understand and relate to. The user, upon understanding the presentation may choose to interact with it for instance 'select a picture gallery to view from the list of galleries presented'. The view handles interaction with the user. However, it does not perform the requested operation for the user; it rather hands over control to the **Controller** by invoking one or more of its operations. Another additional responsibility of the view is to respond to notification of changes in the model (as previously discussed) by updating itself to reflect the current state of the application.

The **Controller** is the coordinator of all activities in the application. As shown in Figure 3.7 above, it is the only part of the architecture that interacts with all the others. The controller can modify the model when necessary; invoke **Services** when needed and coordinate view-to-view relationships. The controller also "acts as Façade for miscellaneous application logic; progress and error reporting; confirmation, validation; security and authentication" (Berkovitz, 2006). The controller is not responsible for communicating with the outside world; this is where the service comes in.

The **Service** is responsible for all remote operations and logic. All communication with the outside world such as Web services, HTTP calls, Remote Procedure Calls, etc are encapsulated by the service. The service also populates the model with remote data.

A brief description of how a component gets created in the space client application will give a better understanding of how the MVCS architecture is applied in the design of the space client. The UML sequence diagram below illustrates the process.

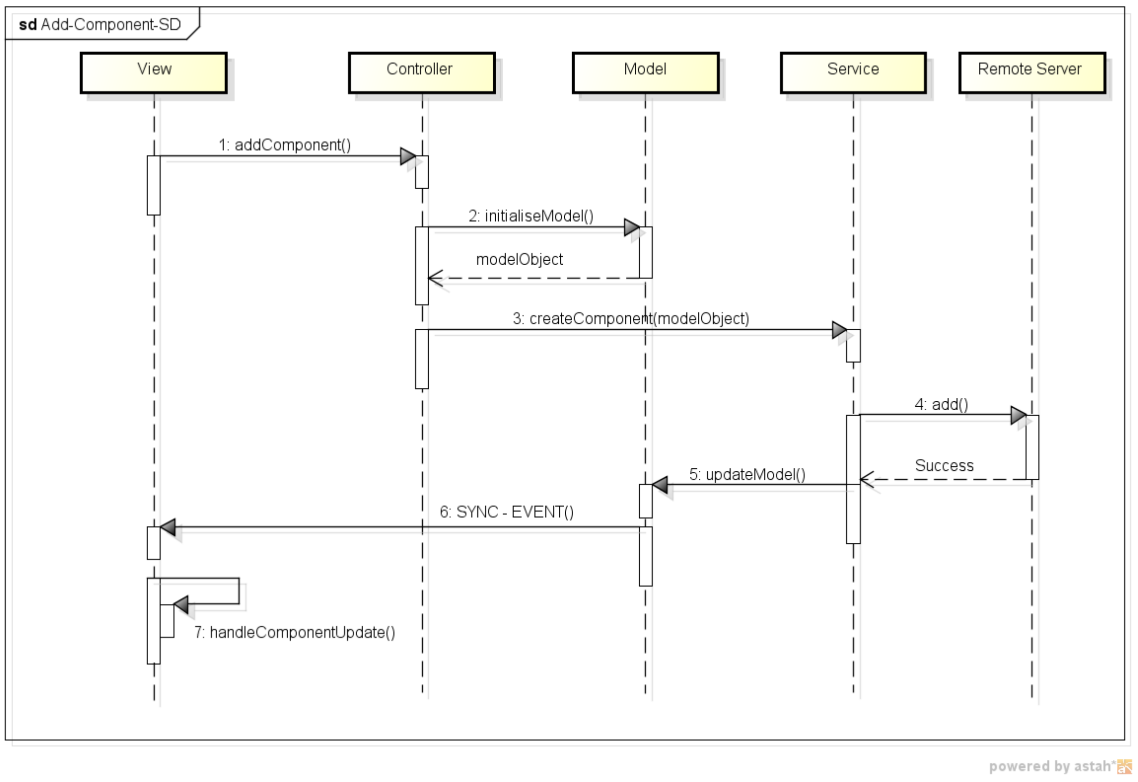


Figure 3.8 - Add media component sequence diagram

Based on presentation layer of the space client described in the space client layout section (see section 3.4.2), the process is as follows:

A. When the user elects to instantiate a component (e.g. by clicking its representation icon on the toolbox), the presentation layer invokes an operation (*addComponent*) on the controller.

B. The controller in turn invokes an operation (*initialiseModel*) on the model that causes the model to initialise default data required by the specified component.
C. The controller then goes ahead to invoke an operation on the service component (*createComponent*) passing the initialized model as the single parameter to the operation.
D. The service makes a remote call (*add*) to the remote server using the NetConnection and the flash.Net.Responder objects. The NetConnection object establishes the link to the remote server so that the remote call can be invoked while the Responder object handles return values from the server related to the success or failure of the remote operation (ASDocs4, 2011).
E. If the remote operation is successful, the model is updated and a SYNC (synchronize) event is dispatched which notifies the presentation (view) of the changes in the model.
F. The view then updates its display components. In the case of a new component, it is then instantiated on the client side and displayed on the whiteboard. For a component that already existed, the state is updated to reflect the current state in the model.

The MVCS architecture helps structure the different parts of the application and organize the relationship between them thus resulting in a highly decoupled system that is easily extensible and scalable. A weak link still exists in the architecture however. This weak link is ‘the object creation logic’. In the next section, this is discussed and a solution proffered to it.

3.5 Inversion of Control and the SWIZ Framework

In object oriented programming, encapsulation is supposed to hide the internal implementation details of an object from users by separating its interface from its implementation. The data and the implementation code for the object should be hidden behind its interface. However, this is not usually the case when creating objects. Consider the code below:

```
MyClass myObject = new MyClass(actual parameters required)
```

Obviously, the class creating this object needs to know the actual parameters required by the constructor to create this object as well as the order in which it is required. Creating objects is therefore considered to be the ‘weak link’ in encapsulation. It can therefore become tricky to decide where (and when) to create objects. The GRASP Creator principle (Larman, 2005) and the gang of four creational patterns (Gamma et al, 1995) suggest how this might be done for example using the Factory design pattern. An alternative approach to this challenge that has gained popularity over the last few years is the ‘**Inversion of Control**’ strategy.

3.5.1 Inversion of Control Strategy

The inversion of control (IoC) design strategy is also known as ‘Dependency Injection’ (DI). DI is “an approach in which a separate object is responsible for populating the fields of other objects with correct implementations, instead of these other objects being responsible themselves” (Eustace, 2009). This approach according to Eustace gives two major benefits:

i. Objects can be decoupled from their implementations by declaring the objects’ fields as interfaces. This is also known as ‘design by contract’.
ii. The object creational logic is separated thus making the purpose of the object clearer.

As a follow up on the second benefit, IoC also solves the weak link challenge described earlier thus enabling our MVCS architecture to be fully decoupled as planned. With IoC, the application objects are instantiated in a separate layer and then passed into the application as needed. Sudgen (2009) emphasises the need to decouple the objects of an application “so they can change independently of one another and be tested in isolation”. According to Sudgen, “applying IoC can make it easier to restructure a user interface, substitute the service integration layer, and refactor the models containing the logic and state of your application, amongst other benefits”.

The concept of decoupling the objects of an application can be further explained using an example. The figure below shows how objects are instantiated in a traditional application.

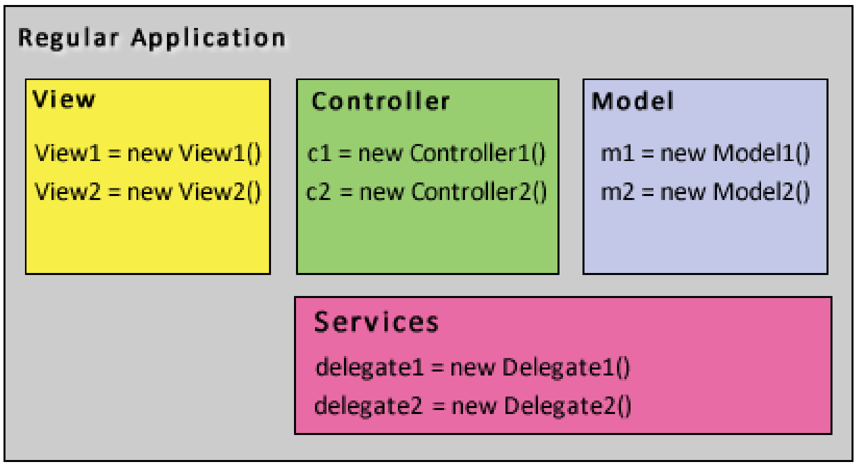


Figure 3.9 - Objects instantiation in a traditional application

(Adapted from Eustace, 2009)

From the figure it can be seen that the architecture is well structured just as described in section 3.4.3 using MVCS. However, in creating the objects, the traditional approach is used thus coupling the different layers to each other once again. In order to understand how IoC solves this, the figure below shows an IoC container handling the creational logic as well as the dependency injection logic.

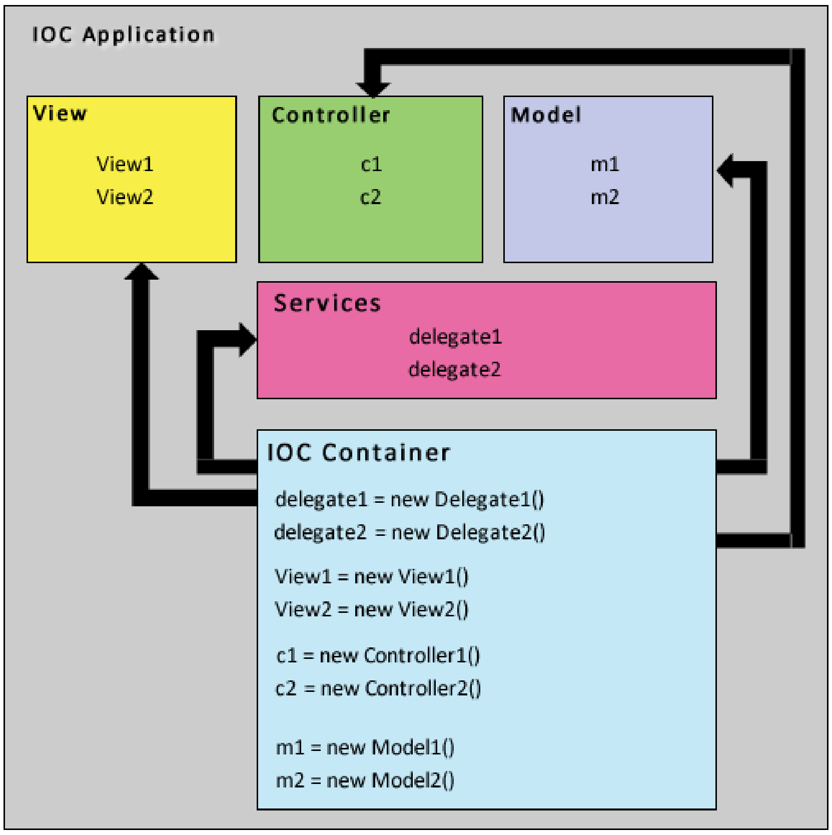


Figure 3.10 - Objects instantiation in an IoC application

(Adapted from Eustace, 2009)

In this case, the required objects are defined for each layer but not instantiated at compile time. At runtime, the IoC container instantiates the required objects for each layer and injects them as needed. This is also known as '**wiring**' the application.

Inversion of control is usually not as easy to implement as described hence the recourse to IoC containers. An IoC container provides a framework for implementing dependency injection in a consistent and declarative way (Eustace, 2009). Examples of such frameworks for Flex include (Eustace, 2009; Sudgen, 2009):

- Spring ActionScript (http://www.herrodius.com/blog/)
- Parsley (http://www.spicefactory.org/)
- Flicc (http://flicc.sourceforge.net/)
- SmartyPants IoC (http://code.google.com/p/smartypants-ioc/)
- Swiz (http://swizframework.org)

The Swiz framework was used in designing the space client. A brief description of the Swiz framework and how it is applied in the design is discussed next.

3.5.2 The SWIZ Framework

Swiz is "a framework for Adobe Flex and ActionScript that aims to bring complete simplicity to RIA development" (Swiz, 2011). It is more than just an IoC framework; it is meant to be a complete solution for Rich Internet Application (RIA) architecture (Eustace, 2009). Swiz provides the following (Swiz, 2011):

- Inversion of Control / Dependency Injection
- Event handing and mediation
- A simple life cycle for asynchronous remote methods
- A framework that is decoupled from the application code

The basic Swiz configuration can be done in three steps

1. Create the beans file
2. Add a SwizConfig to the application and pass in the beans file
3. Add the metadata to the classes into which you want to inject dependencies

Beans in Swiz are MXML (See Section 5.1.1) files that extend the *Swiz BeanLoader* class (Orlando, 2009). The content of a beans file is usually a series of object declarations within the beanloader tags. After loading the Swiz configuration, the objects can be wired to the application either by calling 'Swiz.getBean ("bean-name")' or using metadata such as *Autowire*, *Mediate*, etc. The Autowire metadata tag is used for component wiring, while the Mediate tag is used for dynamic mediation of events during the bubbling phase of the event life cycle (Orlando, 2009).

In the space client application, the Swiz framework is used to inject the dependencies needed and also mediate some events. Controller objects (e.g. whiteboardController), model objects, view objects and service objects are declared in the 'Beans.mxml' swiz beans file. The *SwizConfig* class is then used to inject the beans file into the application. The table below contains a list of some objects managed by the Swiz framework in the space client application.

ARCH. LAYER	**OBJECT**	**OTHER DEPENDENCIES**
CONTROLLER	whiteboardController	whiteboardView, whiteboardService whiteboardComponentFactory
	edgeController	whiteboardView, edgeService whiteboardComponentFactory
MODEL	whiteboardModel, cursorModel, edgeModel	None
VIEW	whiteboardView, whiteboardViewport	whiteboardModel
SERVICE	whiteboardService	whiteboardController
	chatService	chatController
	spaceInfoService	spaceInfoController

Table 3.2 - Some objects managed by Swiz Framework

3.6 Summary

Designing for learning is a task that requires careful consideration at many levels. This chapter has presented a detailed description of the overall design of the system. The system design began with identifying the implicit assumptions so they can be taken into consideration while designing the system. Next, problem analysis was conducted which identified the basic building blocks for the system based on the phases of learning embodied in the literature of learning. The chapter also discusses the architecture of the system which is carefully designed to ensure that the system is easy to extend in the future. The MVCS architectural pattern was chosen to achieve clear separation of responsibilities in the system and also ensure a cohesive but loosely coupled system. To further ensure the system is fully decoupled, the inversion of control strategy (handled by the Swiz framework) was used for object creation and dependency injection. The next chapter focuses on the design of multimedia components in the system.

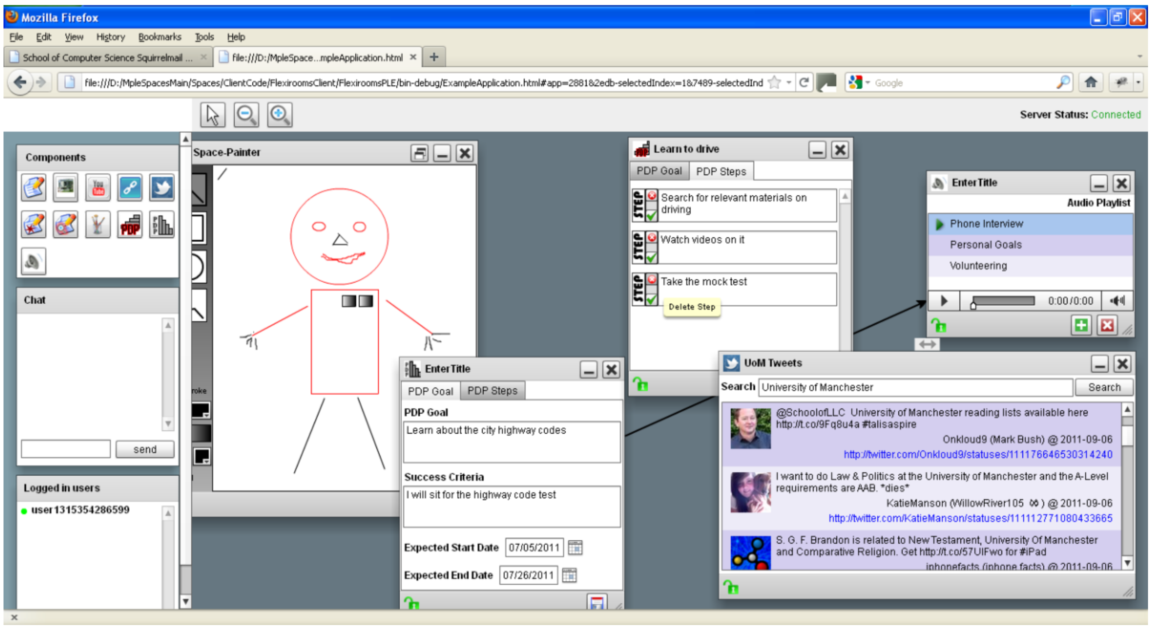
Chapter 4

Component Design

The overall design of the system was discussed in chapter 3; this section describes the design of media components within the system. The following new components were added to the existing ones:

1. The PDP Component
2. The Audio Component
3. The Twitter Search Component
4. The Space Painter Component

A lot of attention is given to the personal development planning component also referred to here as the Personal Development Planner. This is because of its pedagogic importance which in turn made it a major focus of this project. The design of the component is based on the theories of PDP and CPD discussed in section 2.5. The design of other components such as the twitter search component; the audio component; the space painter; etc is also described. The screenshot below shows the finished work.



Search for relevant materials on driving

Watch videos on it

Take the mock test

Phone Interview

Personal Goals

Volunteering

0:00/0:00

Search: University of Manchester

@SchoolofLLC University of Manchester reading lists available here http://t.co/9Fq8u4a #talisaspire
Onkloud9 (Mark Bush) @ 2011-09-06
http://twitter.com/Onkloud9/statuses/111176646530314240

I want to do Law & Politics at the University of Manchester and the A-Level requirements are AAB. *dies*
KatieManson (WillowRiver105 ʚɞ) @ 2011-09-06
http://twitter.com/KatieManson/statuses/111112771080433665

S. G. F. Brandon is related to New Testament, University Of Manchester and Comparative Religion. Get http://t.co/57UIFwo for #iPad
iphonefacts (iphone facts) @ 2011-09-06

PDP Goal: Learn about the city highway codes

Success Criteria: I will sit for the highway code test

Expected Start Date: 07/05/2011

Expected End Date: 07/26/2011

Figure 4.1 - Screenshot of the finished space

4.1 Overview of the Personal Development Planning (PDP) Component

The PDP component enables the learner to take control of his learning. The process begins with a self audit (sometimes referred to as skills audit). Self audit is carried out by the learner with regards to the subject about to be learnt. It involves the learner finding out what he already knows about the subject (or skill) and what is to be learnt. After performing a self audit, the learner proceeds to state his learning goals and success criteria. Steps to be taken to achieve them within a specified time frame are also documented. The completed personal development plan therefore becomes a guide for the learner to organise and document his learning process.

In the media space application, a PDP component can be used document the learning goal, success criteria, steps, etc. Other relevant media components can then be connected to the PDP component to represent various materials linked to the learning goal. As the learner continues to attach other useful components to the PDP component, it evolves into a knowledge artefact that can be (re)used by other learners to achieve similar learning goals. Since all this happens within the media space, it is the content of that space that becomes the knowledge artefact.

Take for example, a learner who wishes to learn how to play the piano. He will select a PDP component and fill in the necessary data such as the goal, success criteria, target time frame, action steps, etc. Having done that, it becomes quite easy for him to search for useful materials that will help him achieve his goal by following the listed steps. Assuming he finds a YouTube video, he can then use it in a video component and link it to the PDP component using one of the edges (directional or otherwise). The edge can be labelled to further clarify the relationship of the linked video with the PDP process. The label also indicates the purpose of the linked video. Other components such as links, text, images, etc can be linked to the PDP component thereby resulting in a knowledge artefact that can be (re)used by the learner (or others) to achieve the intended goal.

Furthermore, a PDP component can also be attached to another PDP component thereby forming a chain of goals and actualisation steps. For example, a PDP component whose goal is ‘Learn to program in PHP’ can be linked to another PDP component whose goal is ‘Learn to develop web applications’. The important data requirements for building the PDP component were gotten by cross comparing different models and approaches in the literature. The next section discusses these in details.

4.1.1 Data Requirements for the PDP Component

The data required for the PDP component are as follows:

- PDP Goal or Learning Goal
- Success criteria
- Target dates
- Steps to be taken
- Status of the process
- Reflection and evaluation notes

The PDP Goal is the main aim of the learning process. The learner sets a goal for himself and then specifies success criteria to be used to ascertain that the goal has been achieved. An example of a goal and a success criterion is given below:

- **PDP GOAL**: Learn to develop web applications in Rails
- **SUCCESS CRITERION**: I will develop a Couples-Matcher application for the School of Computer Science May Ball, 2012

The target dates include target start date and end date. For learning to be effective, the goals need to be **S**pecific, **M**easureable, **A**chievable, **R**ealistic, and **Time bound** (SMART). The last item is the essence of the target dates. Setting target dates ensure that the learner does not spend too much time trying to achieve a learning goal at the expense of other things. It also enables the learner to develop added skills such as time management, ability to learn things quickly, etc.

The steps to be taken are listed out carefully and can be used to determine the status of the process. Finally, reflection helps the learner to think about the process at each stage and also at the end. An evaluation can then be carried out by the learner to improve the process in the future.

Designing the PDP component began with an initial design based on the descriptions above and concepts from the literature discussed in section 2.5. The initial design was then refined until an acceptable design was gotten. This process is discussed next.

4.1.2 Initial Design of the PDP Component

The initial design of the PDP component comprised three multimedia items namely:

1. The main PDP component,
2. The PDP Action List Component, and
3. Other Media Components attached as needed (e.g. Text, Links, Audio, Video, etc)

The illustration below shows the initial conceptual design of the PDP component:

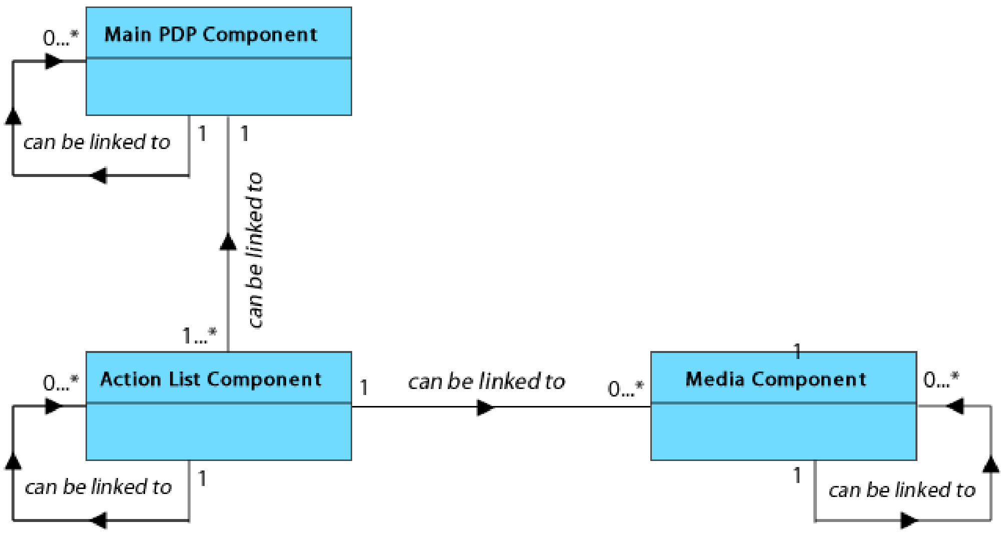


Figure 4.2- Initial conceptual design of the PDP component

The figure shows the relationship between the items that make up the PDP component. The following can be deduced from the diagram:

- The main component can be linked to one or more Action List component(s)
- The Action List component can be linked to zero or more media components
- Each item (the main component, action list component, media components) can be linked to a similar item to create a more complex learning plan.

The figure below shows a sketch of the user interface design for the components described thus far:

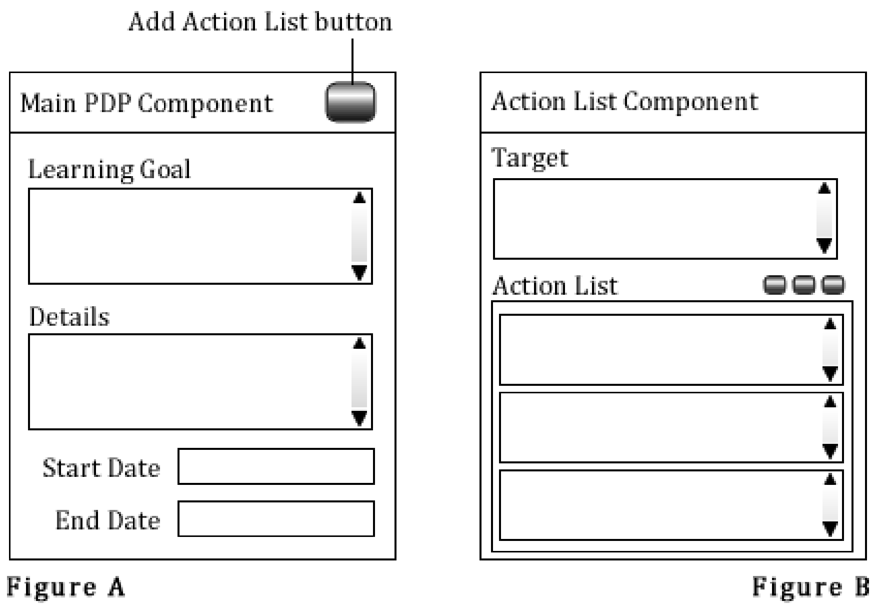


Figure 4.3 - GUI design for the PDP component (A) and Action List Component (B)

In order to use the PDP Component according to this design, the learner will pass through the following stages:

1. **The Main PDP Component**: This will be used to document basic PDP data such as the goal, the success criteria, steps, etc. A button on this main PDP component enables the learner to create another sub-component called 'Action List component' which is automatically attached to the main PDP Component as shown in Figure 4.4 below.
2. **The Action List Component**: This component enables the learner to set a target and then outlines plans to achieve the specified target. According to this design, it should be possible to connect multiple action list components to the main PDP component. Each connected action list component should contain a target that contributes to fulfilling the overall goal recorded in the main PDP component.
3. **The Media Components**: After preparing action list components, the learner can then connect other relevant media components to the action list components such as audio, video, link and text components.

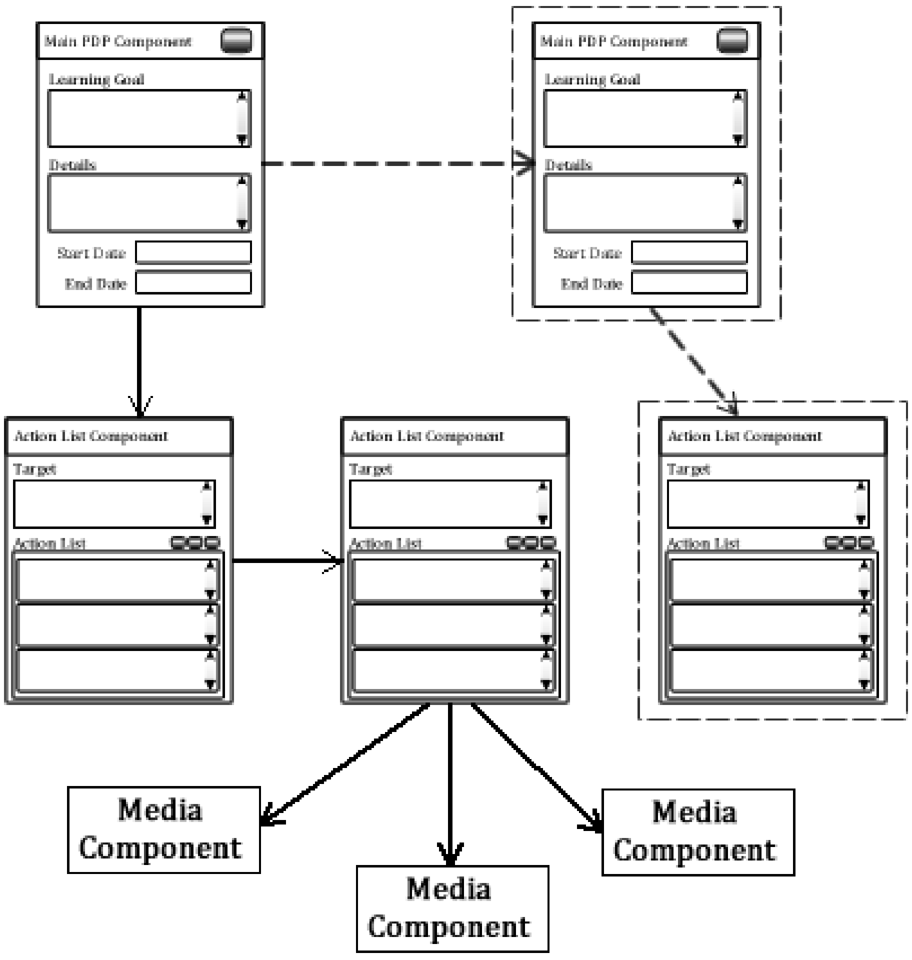


Figure 4.4 - Personal development plan using the initial PDP design

The figure above is an illustration of a possible combination of the different components to create a complete personal development plan. As described in the previous section, each constituent component can be linked to a similar component or another component depending on the complexity of the plan.

The initial design of the PDP component was collaboratively evaluated with my project supervisor during which certain challenges were discovered that led to this design being refined to produce a second design.

4.1.3 Evaluation of the Initial PDP Component Design

The initial design of the PDP component was indeed an attempt to capture all the concepts in the literature on PDP. However, that came with a trade off. Doing so introduced a lot of complexity into the final product. The component was complex, the process of using it was complicated and non intuitive to users. This designed was critiqued and the following impediments were identified:

- The component required a steep learning curve for the user.
- The component (or group of components) takes up too much space on the whiteboard which means the user has to drag the hidden parts to view each time he needs access to them. This can become an exhaustive task and discourage the main purpose which is to learn.
- The complex structure of the component requires a lot of data to be maintained about the individual components and about the relationships among them. This introduces further complexity in the model and network layers in respect of data storage, transfer and synchronisation between collaborative users.
- The complex structure also requires the user to keep track of many components; adding a cognitive overload for the learner. While trying to organise his or her learning materials, s/he is also trying to organise the PDP components. Hiding all other action list components and their sub components when one is selected could be helpful but this would mean introducing additional controls to hide and reveal components, adding to an already steep learning curve.
- In terms of the learning goals and sub-objectives required for each action list component, it was noted that breaking down a learning goal to sub-objectives can be a difficult task. Removing the objectives from the action list control could be helpful in this case but that also means the component becomes just a list of steps for achieving the main goal. This was the first indication that both (main PDP and Action List components) could be merged to form a single component.
- Breaking down a learning goal into multiple objectives can also be time consuming and will discourage the average learner from using the component frequently to support his learning.

These points led to the decision to refine the design so as to make the component more usable. The objective of the refinement was to “simplify the design on the basis that students deal best with very simple things” (van Harmelen, 2011b).

4.1.4 The Refined Design of the PDP Component

After carefully evaluating the initial design and identifying the points above, a series of refinements of the design of the PDP component were made, until a final design was produced that appeared to be more usable than the initial design. The refined design appears in the next figure.

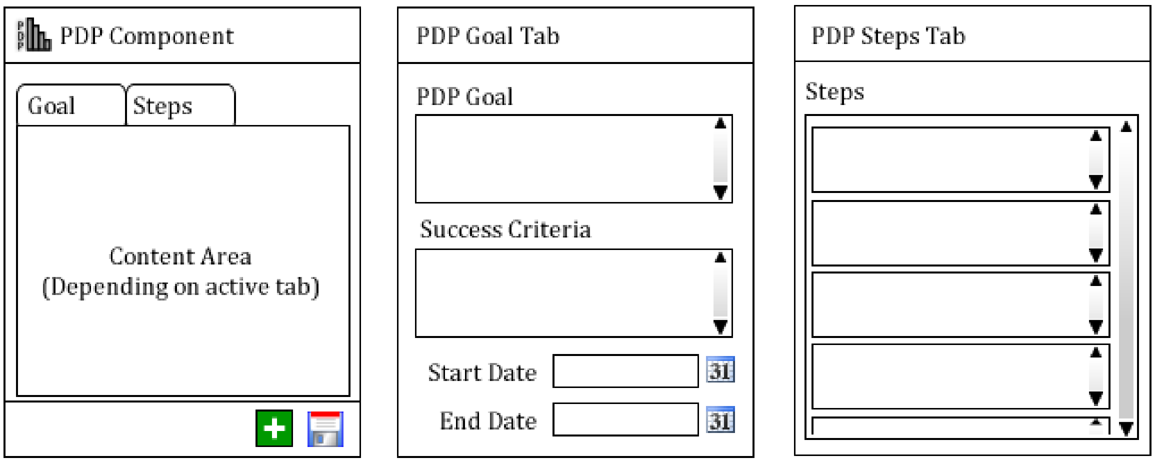


Figure 4.5 - Refined design of the PDP Component

The refined design eliminated the action list component and merged the steps with the main PDP component. According to this design, all the learner is required to do is:

1. State his learning goal
2. State one or more success criteria
3. Specify target start and end dates where necessary
4. List steps to be taken in achieving the specified learning goal
5. Attach other media components as required to document the knowledge necessary to achieve the learning goal

The final design simplifies the concept such that the component can be used by the learner with minimal assistance. In order to achieve a neat user interface layout, a 'tabbed navigator' control was used to separate the PDP Goal from the PDP steps. Further details about the design of the component are discussed next.

4.1.5 Design Details for the Refined PDP Component

The illustration below shows the simplified class diagram for the refined PDP component.

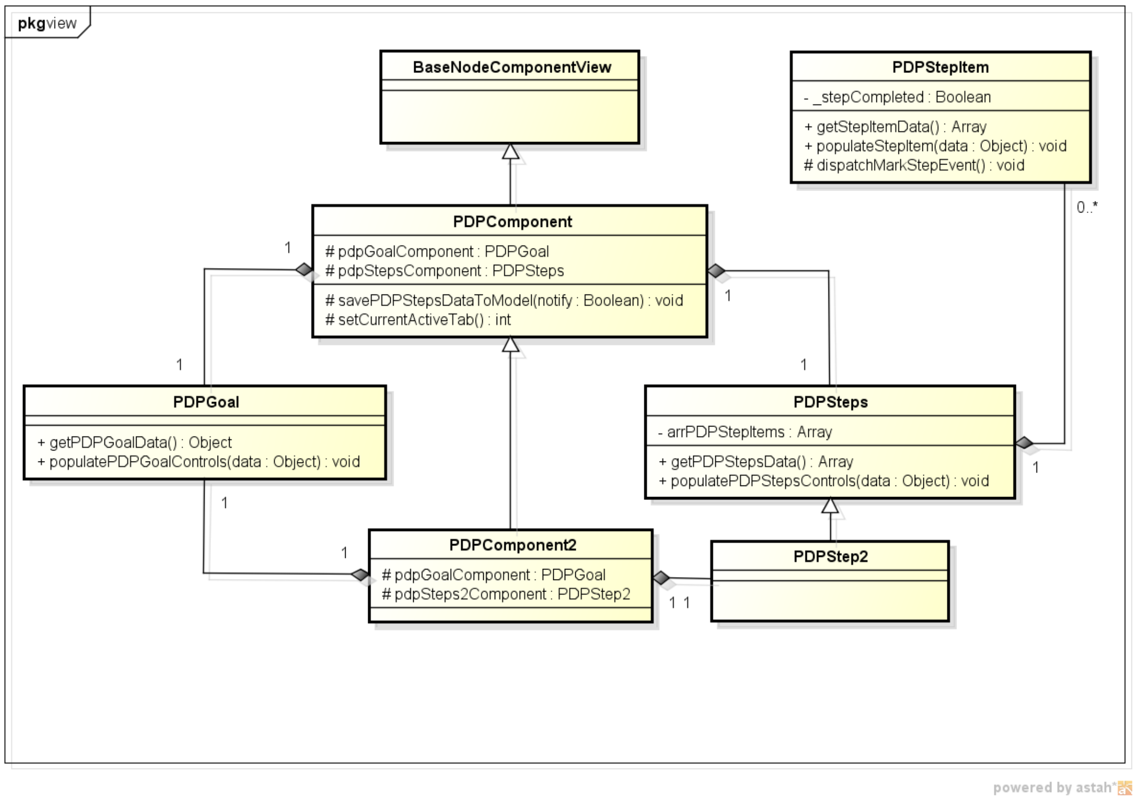

pkgview
BaseNodeComponentView
PDPStepItem
- _stepCompleted : Boolean
+ getStepItemData() : Array
+ populateStepItem(data : Object) : void
# dispatchMarkStepEvent() : void
PDPComponent
# pdpGoalComponent : PDPGoal
# pdpStepsComponent : PDPSteps
# savePDPStepsDataToModel(notify : Boolean) : void
# setCurrentActiveTab() : int
PDPGoal
+ getPDPGoalData() : Object
+ populatePDPGoalControls(data : Object) : void
PDPSteps
- arrPDPStepItems : Array
+ getPDPStepsData() : Array
+ populatePDPStepsControls(data : Object) : void
PDPComponent2
# pdpGoalComponent : PDPGoal
# pdpSteps2Component : PDPStep2
PDPStep2
1
1
1
1
1
1
0..*
1
1 1
powered by astah*

Figure 4.6 - Simplified class diagram for the refined PDP Component

The responsibilities of the component are split among multiple classes. The *PDPGoal* class handles all responsibilities regarding the learning goal, success criteria and target dates. The *PDPSteps* class manages the data and functionalities associated with the steps. This class (*PDPSteps*) is composed of one or more *PDPStepItems* which is another class that encapsulates the functionality of a PDP step for example marking a step as completed or otherwise (step status). The *PDPSteps* class is responsible for adding and removing *PDPStepItems*. When it does this, it dispatches an event to notify all observers (objects listening to the event). The *PDPComponent* class captures this event and invokes a method on the controller to update the model.

The *PDPComponent* class does not function in isolation. It requires the services of other objects in the system such as the controller, the service and the model objects. The diagram below shows the architecture of the PDP Component with regards to other objects in the system.

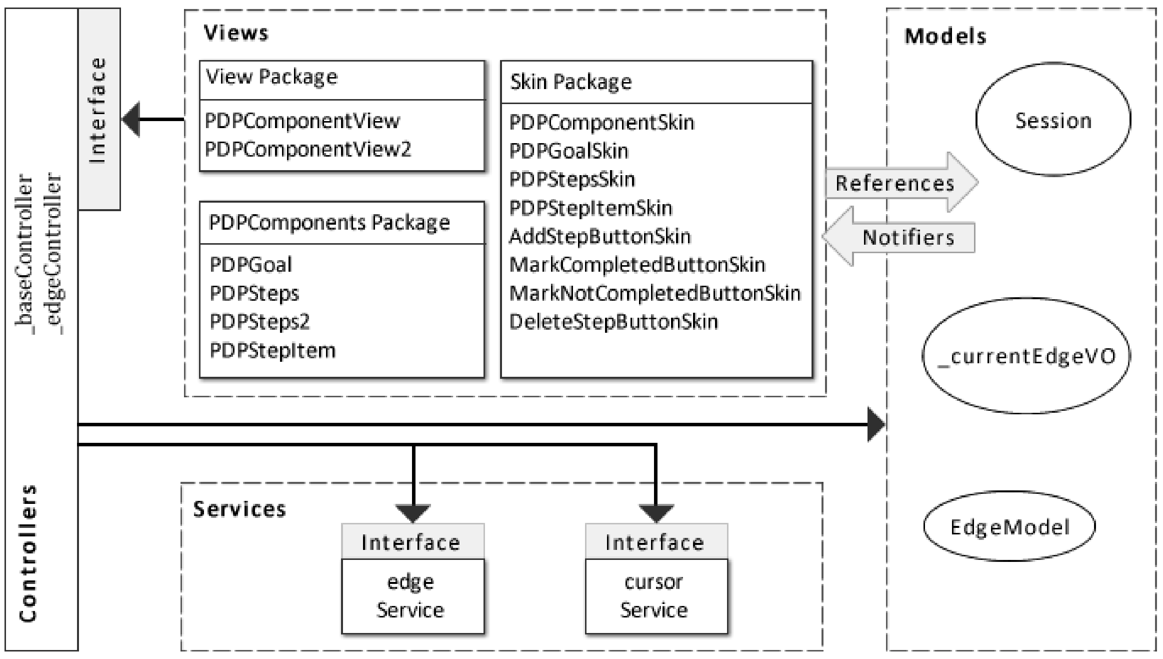


Figure 4.7 - Architectural overview of the PDP Component

The architectural design shown above is based on the MVCS architecture described in section 3.4.3. The view layer is further decomposed into sub layers to separate the styling and skinning of the component from the functionality and interaction logic of the component. Separating a component's display skin from its interaction logic is considered best practice when building components in flex. It allows for loose coupling between the component and its constituent parts thus giving the developer the ability to swap any part for an alternative when needed. This separation of skin parts is also the major difference between Flex 3 components (aka Halo Components) and Flex 4 Components (aka Spark Components). Flex is further discussed in section 5.1.1. The refined PDP component though simplified; maps well to the PDP learning approach discussed below.

4.1.6 Mapping the PDP Learning Approach to the Refined PDP Component

The PDP learning approach is described in section 2.5.1. This approach can be mapped to the PDP component as follows:

PDP LEARNING PHASES	IMPLEMENTAION
Auditing	Not implemented. This is assumed to be done externally by the learner before proceeding to use the PDP component
Planning	PDP Goals Tab for recording goals and success criteria Target dates for time span estimation PDP Steps Tab for brainstorming steps to be taken
Executing	Connecting relevant media components to the PDP component and populating them with useful content
Reflecting	There is no tab on the PDP component for this yet but a Text Component can be connected to the PDP Component and used for reflection notes

4.1.7 Improvements to the Refined Design

Although the refined and accepted design of the PDP component was simplified enough, there still existed room for improvement. One prominent question was "how will learners prefer to manage the steps in the process". This question is important as it can determine which controls should be used in designing the *PDPSteps* component and how they should be used. If the learner prefers to list all the steps together in one place, then that might require a 'TextArea' control. On the other hand, if he prefers to manage each individual step separately, then several 'Textbox' controls might be the better option. To answer this question, two versions of the PDP component were built; one for each style of steps management.

Figure 4.8 below shows the design of the steps component for both versions. The first version requires the user to add each step individually in the steps component provided. The user can add another step component by clicking the add step button. The second version however allows the user to manage the steps in a traditional way by typing them in a single list.

Both versions were evaluated by users and the results are documented in section 6.

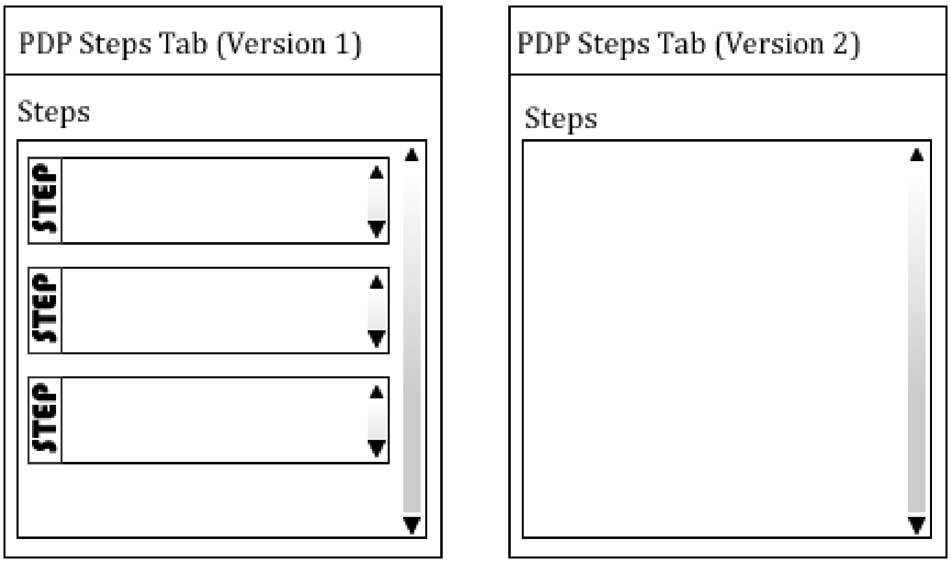


Figure 4.8 - Different versions of the PDP steps component

4.2 The Audio Component

The audio component is used to listen to audio files. These files can be added by specifying their URLs on the internet or alternatively uploading them from the user's computer. The initial design of the component enabled the user to specify a single URL which the component stored and played whenever the user elected to listen to it. This design was reviewed and improved upon. The reason for this was because using the component does not involve any visual interaction between the user and the system apart from simply instantiating and specifying the audio URL for the component. Therefore, listening to more than one audio file means the user will have to instantiate multiple audio components. These components will take up space on the whiteboard and consume memory too. A better design is to enable a single audio component play audio files from different sources. This refined design is illustrated below:

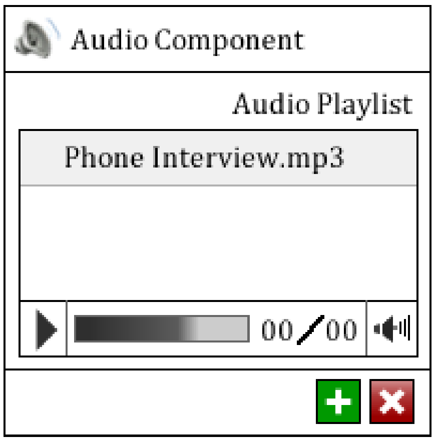


Figure 4.9 - Refined design of the Audio component

The refined design featured a ‘playlist’ which listed all the audio files added to the component. To play the audio file, the user either double clicks one of the listed sources or selects a source and click the play button. The sources of the audio files can be specified by activating the ‘add audio source button’ which brings up the interactive dialogue box pictured below.

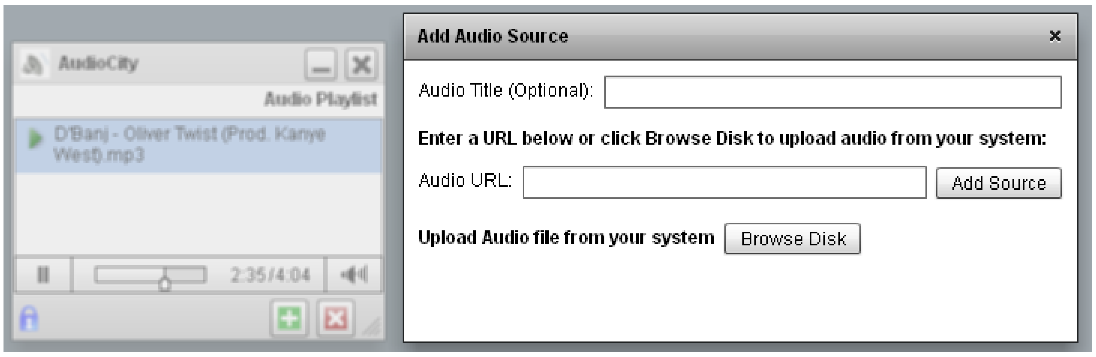

AudioCity
Audio Playlist
D'Banj - Oliver Twist (Prod. Kanye West).mp3
2:35/4:04
Add Audio Source
Audio Title (Optional):
Enter a URL below or click Browse Disk to upload audio from your system:
Audio URL:
Add Source
Upload Audio file from your system
Browse Disk

Figure 4.10 - Add audio source dialogue box

The dialogue box allows the user to either specify an audio URL from the internet or select an audio file from his system which is then uploaded and added to the playlist. The user can optionally specify a title for the audio source. Where a title is not specified, the component detects the title from the source and displays it in the playlist. Audio sources can also be deleted from the component by selecting them on the playlist and activating the delete source button.

4.3 The Twitter Search Component

Twitter is a popular online social networking and micro-blogging service. Users of the service can post, read and re-post text based information of up to 140 characters. These posts are informally known as "tweets". The twitter component enables the user to search for keywords within tweets. The user specifies a search keyword and the component retrieves the result of the search and displays them to the user. Twitter can be a very good source of information because users usually insert shortened URLs into tweets which when clicked points the reader to more details about the tweet. Additionally, the user can stay current about trends on a particular topic from the results returned since they are listed by date; the most recent ones first. The illustration below shows the design of the twitter search component.

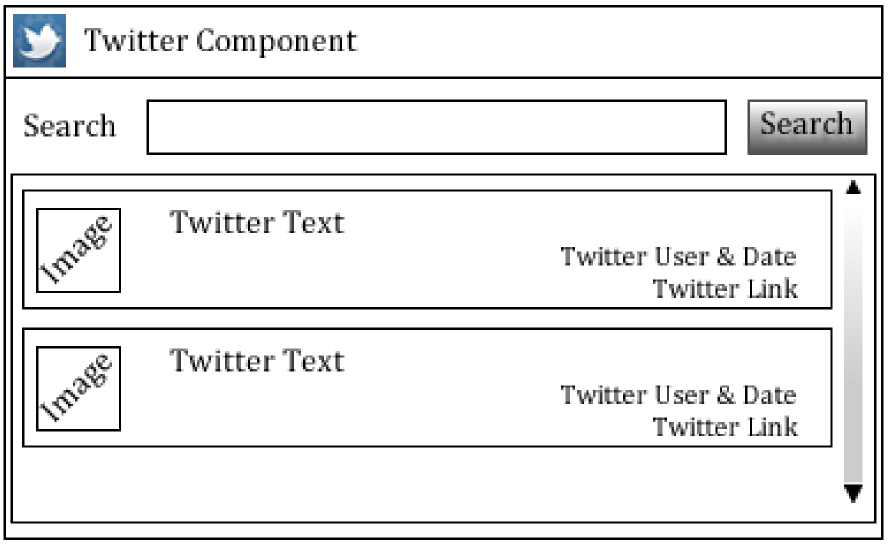


Figure 4.11 - GUI design of the Twitter search component

4.4 The Space Painter Component

The space painter component is a drawing component. This component enables the user to use basic shapes and colours to draw graphics within the application. The component was adapted from the shapely demo application by Haase (2010) in his book ‘Flex for fun’. The component comprises 2 major sub components namely:

1. The toolbox and
2. The drawing canvas

An illustration of the graphical user interface design of the component is shown below.

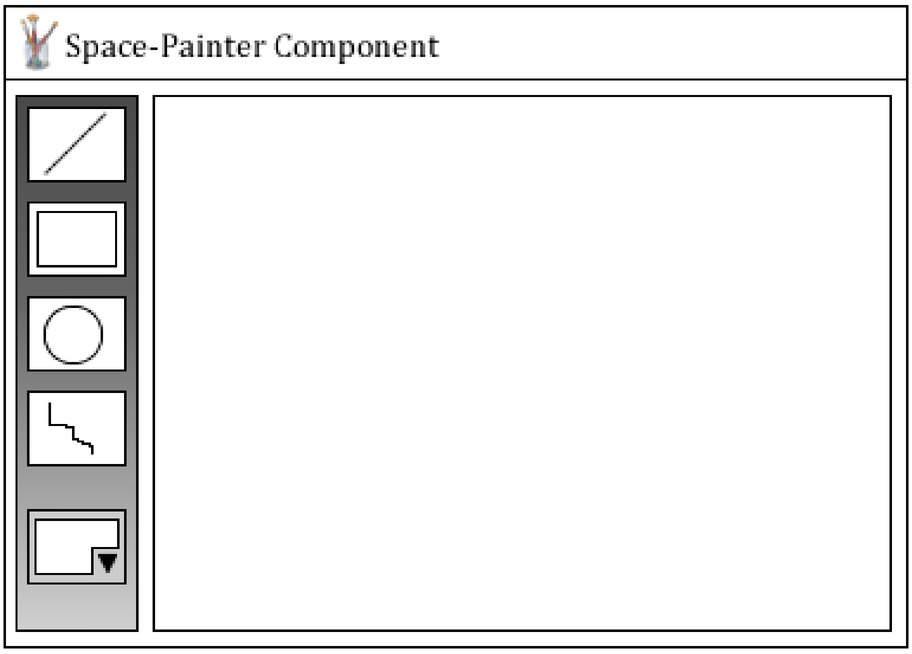


Figure 4.12 - GUI design of the Space-Painter component

To use the component, the user has to select a tool such as rectangle, ellipse, line, etc from the toolbox. The fill and stroke colour section of the tool box allows the user to specify line and fill colours for the shapes to be drawn. The user can also give the shape a gradient fill by specifying two colours as the fill property.

4.5 Summary

After describing the design of the overall system in the previous chapter, this chapter focused on the individual components design. The design of the PDP component, audio component, twitter search component and space-painter component have been fully covered in this chapter. For each component, the necessary data requirements were first identified then, an initial prototype designed and evaluated. The evaluation of the initial design highlighted some aspects which were then improved to produce the final design of the component. The next chapter discusses the implementation details.

Chapter 5

Implementation

This section discusses the implementation of the system. The tools used for implementation are identified and discussed. The implementation process is also fully discussed. Finally, challenges faced during the implementation and how they were solved is also explained.

5.1 Implementation Platform, Languages and Tools

The multimedia learning space is design to be accessed from the internet via a link on the PLE. However, the application is designed to be a Rich Internet Applications (RIA). A Rich Internet Application is web application that feels, looks like and functions like a desktop application. A traditional web application is stateless. This means that the state of the connection with the server is not maintained; rather every new request from the user is treated as a new and independent transaction. Also, in a traditional web application, most of the processing tasks are performed on the server side. With RIAs, the reverse is the case. The connection state is maintained and most of the processing is moved to the client side with the application contacting the web server only when it is necessary. This minimises the amount of round-trip to the web server and therefore increases the response time of the application. JavaFX, AJAX, Adobe Flex and Microsoft Silverlight are examples of some technologies used to build RIAs. Adobe Flex was the technology of choice for implementing the multimedia learning spaces.

5.1.1 Adobe Flex Framework and Flex SDK

Flex "is a highly productive, free, open source framework for building expressive mobile, web, and desktop applications" (Adobe, 2011). Web and mobile applications built using Flex share the same code base and can both be deployed as desktop applications. Applications are built in Flex using MXML tags and Action Script. The codes are then compiled into Shockwave Flash (SWF) files which can then be executed in the Adobe Flash runtime mostly for web environments and the Adobe Air runtime for desktop applications. The Flash runtime is widely available on all major internet browsers while, the Air runtime can be easily downloaded online. Flex applications running in web pages do not require a page reload to update information on the user interface. They connect to remote server side applications to retrieve data which are used to update the client interface as required. These features make Flex a very suitable candidate for building RIAs.

The Flex framework “provides the declarative language, application services, components, and data connectivity developers need to rapidly build rich Internet applications (RIAs) for mobile, web, or desktop” (Adobe, 2011d). Since it is an open source framework, it can be extended to suit the specific needs of the application being developed. The declarative language being referred to by Adobe above is ‘MXML’. MXML is a “declarative XML-based language, used to describe user interface layout and behaviours” (Adobe, 2011d). MXML is used to describe the visual aspects of the application while ActionScript is used for the more programmatic functionality of the application, like the business logic, the client logic, etc (Adobe, 2011d; Haase, 2010). This separation of responsibility in the Flex framework enables the developer to customise the look and feel of Flex components through their skins. Component skins are written in MXML files. They define the graphical elements that describe the visual appearance of the component (Haase, 2010). Additionally, the separation of concern encourages and supports parallel application development by developers and designers. The framework also comes with prebuilt application services and components that help developers build applications faster (Adobe, 2011d). DataGrids, Charts, Formatters, Validators, and other UI controls are some examples of components that come with the framework. Some prebuilt services include data binding service, drag-and-drop, the display system (for managing layout of the User Interface), the effects and animation system, the style system (for managing the look and feel of the controls and components) and the pop-up manager (Adobe, 2011d).

The Flex software development kit (SDK) is a developer toolkit for development in Flex. It includes a compiler, a debugger and a profiler. Each of these tools can be used from the command line (terminal, console) or from an integrated development environment (IDE). The current version of the SDK released by Adobe is version 4.5.

5.1.2 Adobe ActionScript

Adobe ActionScript is an object-oriented language based on industry-standard ECMAScript. When developing applications using Flex, ActionScript is the language used to build client-side application logic (Adobe, 2011). Action script looks a lot like JavaScript in syntax and semantics. It was originally designed by Macromedia for website animation and released with Flash 4. The latest version of the language which is version 3.0 has been expanded to include a lot more functionality including database access. ActionScript is an event based language. This means that actions are triggered by events coming from either the framework or from user interaction with the application.

5.1.3 Adobe Flash Builder

Flash Builder is an integrated development environment for building Flex applications. It was formerly known as Flex Builder and is based on the Eclipse™ IDE Framework. A plug-in also exists that can be installed on any existing Eclipse™ based IDE. Flash Builder's intelligent code assist function helps accelerate development in flex. The IDE also has an interactive debugger and profiler which can connect to Adobe Flash Player (debugger version) to assist the developer debug and profile the application. The premium version of the IDE also features a Network Monitor that can be used to monitor and analyse requests sent to the web server and responses received. The latest version of Flash Builder is version 4.5 and it has support for editing MXML, ActionScript and Cascading Style Sheets (CSS).

5.1.4 Version Control System (VCS)

Version Control also known as revision control or source control is a way of managing changes to documents, program source codes or other files. Source control is very useful for tracking changes to files especially in a collaborative environment where more than one person may make changes to a file at the same time. Changes are usually marked by a unique code. The file or group of files being managed can be "branched". This involves making a duplicate copy of the original document and continuing work on the copy while the original remains untouched. The version control system used for this project is called **Git**. Git is a "free & open source, distributed version control system designed to handle everything from small to very large projects with speed and efficiency" (http://git-scm.com).

5.1.5 Astah UML Modelling Tool

Astah is a UML modelling tool. The community edition is free and offers a range of facilities for UML modelling. The Astah UML tool was used to design the conceptual models of the system; the system architecture and package organisation as well as the class diagrams.

5.2 Preparation for Implementation

Extending an existing code base to add new functionalities requires an understanding of the existing code. If the existing code was written using Test Driven Development (TDD), then 'unit tests' will exist and that can be very helpful. After making changes to the code, existing tests can be executed to ascertain that nothing was broken by the change. Where there are no tests, one must do a bit of code reading to understand the structure of the code and how different parts work together in the application before attempting to make changes to the code. In the case of this project, intense code reading had to be done first before the actual implementation due to the absence of unit tests.

5.2.1 Understanding and Improving the Existing Implementation

Understanding the existing implementation involved code reading. However, code reading without an aim can lead to the process becoming an endless journey. In order to avoid this, the task of fixing existing bugs in the current implementation was set as the goal of the code reading process. This provided a direction, focus and the ability to monitor the process carefully. Some improvements that were made to the existing implementation are discussed below.

I. <u>Auto Save in Text Component</u>: The existing text component required the user to click on the 'save' button after making changes for the changes to be updated. The improvement to this required the component to automatically save changes made by the user. This was successfully implemented.

II. <u>Title Disappearance Bug</u>: The title bar of components contains a label that can be edited for each component. An existing bug in the implementation prevented the edited label from being persisted in some scenarios. In the first scenario, when the user double clicked on the title bar to edit the title, if nothing is entered into the textbox, the title disappears. This is not correct as the previous title should have been maintained. Secondly, while editing the title, if the textbox loses focus, the title disappears. Finally, even when the title was persisted, refreshing the browser to reload the application still led to the disappearance of the component title. All these bugs were fixed and the title now persists as required.

III. <u>Fully Expandable Components</u>: Existing components could only be minimised, restored or resized by dragging subject to minimum and maximum dimensions. The desired improvement was for components to be able to expand to fit the entire dimensions of the viewport. This was successfully implemented and tested. A base component that handles the functionality was then implemented such that other components that require this functionality can simply extend the base component by inheritance.

IV. Resizing bug in Video Component: resizing the video component when a video is loaded made it lose its aspect ratio thereby making the video look stretched and out of proportion. This bug is due to the fact that the loaded video is an SWF file and therefore tries to adjust itself when its container's dimension changes. Fixing this bug required that the resizing process be strictly controlled to be in synchronisation with the aspect ratio of the loaded video. Unfortunately, this bug is yet to be corrected as at the time of this report.

5.3 Adding New Components

The improvements and bug fixes made to the existing implementation provided the understanding necessary for adding new components to the application. The following new components were added:

I. The PDP Component
II. The Audio Component
III. The Twitter Search Component
IV. The Space Painter Component

The design of these components is described in section 4. Screen shots of them can be seen in Appendix A. The diagram below shows the generic class diagram for the components:

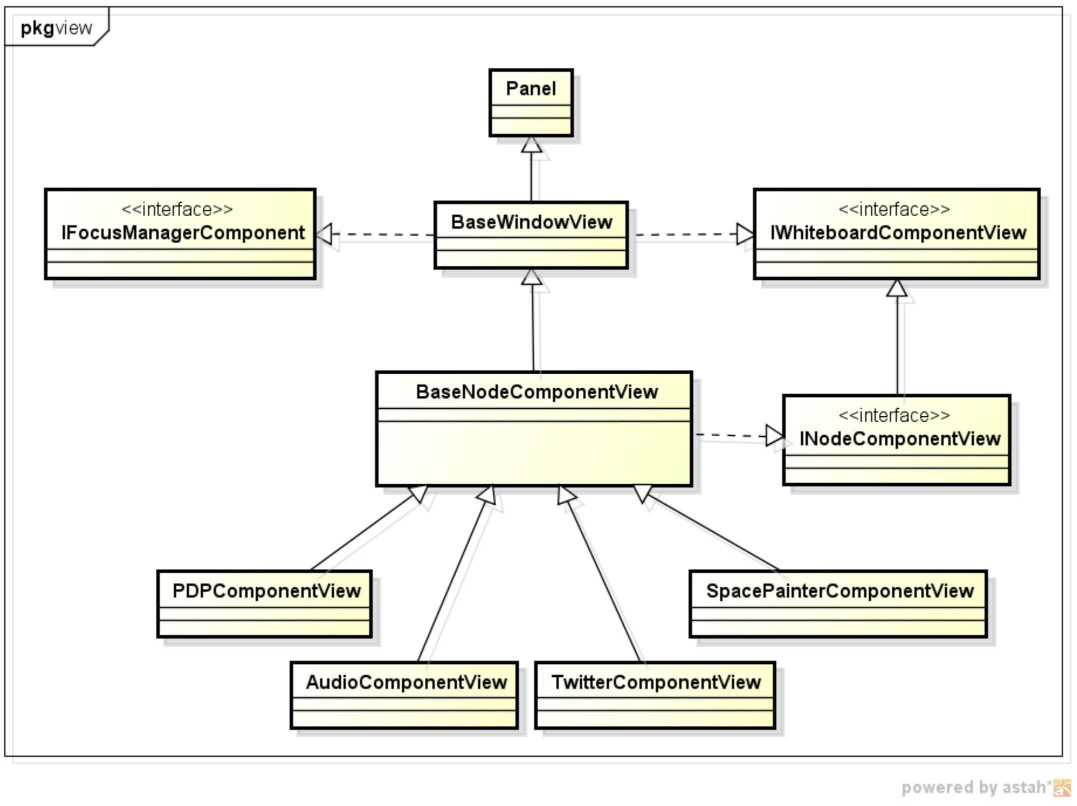


Figure 5.1 - Generic class diagram for the components

The properties and functionalities of each class and interface in the diagram are briefly discussed below.

The BaseWindowView Class

The *BaseWindowView* is the base ActionScript class for all media components in the application. This class extends the Adobe spark panel component and provides the following functionalities for sub classes.

- A titled GUI window which it inherits from the spark Panel class.
- Basic window controls such as the title bar icon; the window resizing icons; the close icon and the status bar
- Custom window events such as minimise and maximise events. These events are dispatched when the user triggers the corresponding actions.
- Custom 'LOCKING' events such as 'Locked by current user', 'Locked by another user', 'Locked by no one'. Locking is used in collaborative mode to control access to shared resources. When one user is using a component that

requires locking, it is obtained through the controller and the appropriate lock events are dispatched.

- Custom effects that get executed when a component is resized, minimised, maximised or moved.
- Implementation of basic functionalities such as ‘minimise’, ‘maximise’, ‘expand’, ‘resize’, ‘close’, etc which are later overridden and customised by sub-classes.
- A reference to the controller and model

This class is also responsible for initialising the basic properties of a component when it is instantiated. These properties include the component title, x-position, y-position, width, height, minimum width, minimum height, maximum width and maximum height. The initial values for these properties are requested from the model and stored locally as properties of the component.

The class also sets up a ‘Watcher’ for the basic properties of the component. A ‘Watcher’ is a function that is executed when the model notifies the component of changes to the property being watched. This is a classical implementation of the observer pattern. The components listen for notification events from the model about changes to specified properties. When this notification is received, corresponding functions are performed for each property that is being ‘watched’.

The BaseNodeComponentView Class

The *BaseNodeComponentView* class extends the *BaseWindowView* class. This class bears the responsibility for the following functionalities:

- Setting up the edge controller which handles the linking of components to each other
- Setting up the title display label and text box which is used to edit the title
- Basic functions for handling the linking of components using edges
- Functions for accessing some basic component properties such as ‘x’, ‘y’, ‘width’ and ‘height’

The Swiz component which was discussed in section 3.5.2 handles the creation of several objects required by the component such as the edge controller, edge view, and edge model. The *BaseNodeComponentView* is also responsible for calling the Swiz component ‘autowire’ method which does the creation and setup of the required objects.

The Implemented Interfaces

Three interfaces can be identified from the diagram namely:

1. IFocusManagerComponent
2. IWhiteboardComponentView
3. INodeComponentView

Two of these interfaces are implemented by the *BaseWindowView* class while one is implemented by the *BaseNodeComponentView* class.

The *INodeComponentView* interface defines four methods used to retrieve the basic properties of the component. These include the 'x' and 'y' position as well as the 'width' and 'height' of the component.

The *IFocusManagerComponent* interface is a Flex interface within the mx package. It "defines the interface that focusable components must implement in order to receive focus from the FocusManager" (ASDocs4, 2011). This interface is implemented by the BaseWindowView class and is used to capture focus for the media components. One practical use for this focus capturing is to control the layering of components. For example if there are four components on the whiteboard and a user selects one of the components that is slightly covered by other ones, the selected component is moved to the top and the other components arranged behind it. The selected component is thus said to have received focus.

The *IWhiteboardComponentView* interface defines the interface for setting the component controller and model. It also defines the interface for initialising the model; binding it to the component and retrieving a reference to it when required.

The Media Components

The media components are the final items in the UML diagram above. They are sub classes of the BaseNodeComponentView class. Functionalities specific to each component are implemented in the components separately while shared functionalities are inherited from the hierarchy. These shared functionalities can also be overridden to customise their implementation for each component. A good example of such overriding is the component 'resize' function which can be overridden to ensure that the components do not get resized beyond a particular visual limit or to ensure they are adjusted evenly based on a specified aspect ratio.

Each component is made up at least two files:

- The View file (ActionScript class)
- The Skin file (MXML file)

Both files make up the ‘View’ layer in the MVCS architecture (see section 3.4.3). The Flex controls or custom controls that make up the visual appearance of the component are defined using MXML in the skin files. Various display properties are set to arrange the controls according to the design required. Flex also provides containers for laying out these controls as desired. The logic for interaction is however moved to the view class file written in ActionScript. This class contains the codes that specify which controls are displayed to the user and when. When the user interacts with the controls, the response is also handled by this class. In Flex programming, implementing a loosely coupled component model is considered best practice. Separating the view logic and content from their visual implementation enables this practice at the view layer.

5.4 General Challenges Faced During Implementation

While there were many challenges faced during the implementation phase, persisting objects was particularly difficult. Initially, the data required to setup each component were meant to be stored in custom property classes for each component. However, the persistence layer posed challenges to this because after successfully persisting the object, they could not be recovered again to set up the component when the application was restarted. A careful investigation showed that this was caused by object references which got lost during model updates. This occurred because ActionScript passes objects by reference only. The references to these objects are broken during the model updates and this leads to loss of data. One way to overcome this is to pass the objects meant for persistence to the model layer ‘by value’.

To pass an object by value in ActionScript, one must find a way to clone the object. Cloning is also not well implemented in ActionScript. Therefore, in order to clone properly; a function that transfers the properties of an existing object to a newly created object must be implemented. However, this does not still resolve the persistence problem because during such cloning, if a property of the former object is an object itself, copying the data still maintains a reference to the former object. Eventually, the concept of persisting data using objects had to be abandoned and a safer approach was taken. The alternative approach involved persisting data as arrays.

5.5 Challenges Faced in Audio Component Implementation

Implementing the audio component required some special measures in dealing with certain changes. Generally, in a collaborative environment, a change made by one user has to be updated across other users' interfaces. For the audio component however, not every change had to be updated for all users. For instance, it should be possible for the users to listen to different audio files at the same time. This particular scenario is very interesting because an audio source being listened to by one user can be deleted by another user and the deletion has to be updated across all users. This update requires that the audio file stops playing and is removed from the playlist. This was initially tricky to achieve as the audio file kept on playing even when the source was removed. It was later achieved however, by maintaining a record of the currently playing audio file and then using this record to search for and stop the audio file when deleted before removing the source.

Another interesting challenge was displaying a 'now playing icon' for one user on a particular audio source on the playlist while displaying the icon for a different source on the playlist for another user. Initially, updating a field to specify the source now playing caused the updated change to display multiple 'now playing icons' for several audio sources in other users' playlists. This functionality was later achieved by managing the currently playing source locally for each user. Thus when an update changes the icon of the currently playing source, it can be restored using the local data source of the user.

The audio component also had some limitations including the maximum file size that can be uploaded and the file type that can be handled by the player. The maximum file size is determined by configurations on the web server processing the upload. This challenge was solved by editing the configuration files appropriately. With regards to the file type, a filter was implemented to ensure that only the types that the player can handle are uploaded.

Another limitation is that internet sources cannot be guaranteed to remain there for as long as the user needs them. The audio component however can inform the user when the source of the audio file is no longer valid. The user can then delete the item or change its source.

A final challenge, which is not solvable here, is that some networks limit upload speed and streaming capability. This is not a problem that is soluble here, but is also one that does not manifest itself much in developed countries.

5.6 Summary

This chapter has described the implementation of the designs covered in chapters 3 and 4. The discussion started with the implementation platforms, languages and tools. A brief discussion of how preparation was made for implementing the new designs followed. Improvements made to the existing implementation were described giving reasons why they were necessary. This was followed by a description of how new components were added to the system. The chapter rounds off with challenges faced during the implementation phase and how these were handled.

The next chapter discusses how the new system was evaluated.

Chapter 6

Evaluation and Analysis

This chapter describes the evaluation and analysis of the system. Although formative evaluation was done throughout the process, a user evaluation was still conducted after the designs were implemented. Details of the evaluation method, process, tasks, participants and results are given in this section.

6.1 Evaluation Method

Cooperative Evaluation is a procedure for obtaining data about problems experienced when working with a software product, so that changes can be made to improve the product (Monk et al, 1993). Cooperative evaluation is a variant of think aloud evaluation. During a 'think aloud' evaluation session, the user performs a number of tasks and while doing that, 'thinks aloud' (i.e. explains and discusses with self aloud while performing the actions) explaining what he is doing at each stage of the task. The cooperative evaluation technique encourages design teams and users to collaboratively identify usability issues and their solutions. By encouraging the user to see himself as a collaborator in the evaluation rather than a subject, a more effective form of evaluation is achieved, thus increasing the utility of the data about problems experienced when working with the product. It is very important to note that in cooperative evaluation, the evaluation subject is the product not the user.

The main activities involved in preparing and running a cooperative evaluation session is as follows:

1. Recruit users
2. Prepare tasks
3. Interact and record

The process begins with recruiting users. The users evaluating the product should be representative of the target population that will eventually use the product. This ensures that useful information is gathered from the evaluation. The number of users that will evaluate the product is also determined. Tasks to be performed by the users are prepared. Selecting the right tasks is crucial for the success of the evaluation. Some points to consider when preparing the tasks are as follows (Monk et al, 1993):

- The tasks should be specific. For instance, ‘perform the normal tasks you usually do with this system’ is not a specific task. ‘Select a rectangle tool, drag across the screen to draw a box’ is a specific task.
- The tasks should be representative of the real tasks users of the product will perform
- The tasks should explore the prototype thoroughly
- The tasks should be doable by the users taking part in the session. This also implies the tasks should be clear enough to be understood by them.
- The tasks should be time bound i.e. there should be an estimated time of completion for each task.

The evaluator can also prepare extra tasks that can further explore a part of the product. These can be useful if the user finishes the specified tasks before the allocated time and extra data is desirable.

The last session involves the tester interacting with the subjects and recording information. This ‘interact and record’ session is divided into four parts:

1. Before the users arrive
2. When the users arrive (before starting the tasks)
3. While the users are using the system
4. Debriefing the users

Before the users arrive, everything should be put in place and fully functional. The prototype to be evaluated should be set up in a preferably quiet environment. The tasks to be performed should be printed and ready for use. Some means of recording the session such as a video recorder should be set up and tested. In the absence of a video recorder, a notebook can be used. The questions to be asked during the debriefing session should be prepared. If questionnaires are to be used, they should also be ready and kept close by.

When the users arrive, they should be put at ease before beginning the evaluation. The session is meant to be conducted in an informal manner; as such the users should be encouraged to see themselves as co-evaluators not as experimental subjects.

While the users are using the system, they should be kept talking. The evaluator must ensure he knows what is happening at every stage and should prompt the user to ‘think aloud' if he is silent for a while. Questions such as the following can be used to prompt the user to speak up during the tasks (Monk et al, 1993):

- How do we do that?
- What do you want to do?
- What will happen if.....?
- What has the system done now?
- What is the system trying to tell you with this message?
- Why has the system done that?
- What were you expecting to happen then?
- What are you doing now?
- What were you expecting to happen then?

If the user gets stuck on a task, the evaluator should assist the user; finding out what the impediment to progress is, or encourage him to proceed to another task.

When the user has finished the tasks, the debriefing session begins. This session can also be recorded for analysis. The purpose of this session is to get more feedback from the user. Apart from discussing about the product that was evaluated, the evaluation process can also be discussed. The evaluator can ask the user questions such as:

- What do you think was the best and worst part of the product?
- What needs to be changed or improved upon urgently?
- How easy did you find the tasks?
- What questions do you have about the product?

Some very interesting comments usually emerge from the debriefing session hence the need to record the session.

Cooperative evaluation is inexpensive and can be done anywhere provided the required resources (the prototype to be tested, the evaluation tasks and recording materials) are present. The results of cooperative evaluation are always very useful because users are a good source of potential design solutions, and when users consider themselves co-designers or co-developers useful responses are elicited from them.

6.2 Participants, Tasks Requirements and Ethics

Cooperative Evaluation can be tasking and is not something that should be repeated. It is better to get it right the first time. Selecting participants and preparing the tasks are important parts of the process that must not be under-emphasised. Furthermore, an evaluation that involves human participants is required to conform to some ethics to protect the interests (and sometimes, but not in this case, the health and safety) of the participants.

6.2.1 The Participants and Task Requirements

The participants for the evaluation were drawn from different areas of studies such as the Business school, Engineering, and the Humanities. The number of participants from computer science was minimised due to their technical background. A total of 10 people comprising 7 female and 3 male postgraduate students participated in cooperative evaluation of the software.

A set of representative tasks were prepared for them to perform after which a questionnaire was given to them to fill. The list of tasks as well as the questionnaires can be seen in Appendix B. The evaluation generally lasted between 20 - 30 minutes per participant and was carried out in the PEVE lab in Kilburn Building, School of Computer Sciences which was already appropriately set up to meet the requirements of the method.

The tasks that were given to the participants to perform required some form of input from the participants. In order to ensure that the participants spend time evaluating the system rather than thinking of what to input, some data was provided for each participant. Each task also had an estimated completion time to ensure that all the tasks fit in the target 20 minutes estimated for the whole session per participant. The “time on task” (TOT) is the time required to complete a given scenario or task. The TOT was carefully set for each task to ensure 100% completion rate for the whole session. However, this turned out not to be so. Some participants had un-anticipated difficulties during the process leading to some of the users taking longer than the estimated TOT.

6.2.2 Research Ethics

Research Ethics are “very simple rules of conduct, recognised by certain organisations, which can be applied to aspects of Computer Science evaluation” (Ethics-CS-UoM, 2010). Research ethics ensure that certain ethical guidelines are followed in conducting the evaluation. They also ensure that the evaluation is useful and not a waste of the participants’ time. The evaluation sessions were required to adhere to the following ethical guidelines:

- The performance of any participant must not be individually attributable.
- The focus of the evaluation must be on the software and not the knowledge or performance of the participant
- The participant's name should not be used in reference outside the evaluation session.
- The participant can leave at any time during the evaluation session if s/he deems it necessary to do so.
- The participant should willingly give his/her consent with regards to the method to be used in recording session.
- Where the session is recorded, the recording must be used only for the purpose of the research and nothing else.

A consent form was given to each participant at the beginning of the evaluation session detailing the ethical guidelines of the session. Each participant was also required to give consent to the form of recording allowed during the session. The forms of recording made available for the session included

- Pen and paper
- Computer screen capturing
- Audio only recording
- Video only recording
- Audio and Video recording

A blank ethical consent form can also be seen in Appendix C.

6.3 Evaluation Results

The results of the evaluation sessions for the individual user tests as well as the simultaneous user tests are described below.

6.3.1 General Results

- All the participants agreed that it was easy for them to find their way around the application. Three participants said “the application was self explanatory”. When asked to expatiate, they explained that since the application is consistent with the way components are represented, it is easy to deduce the usage of other components after using one component. One participant remarked that she is up to 75% confident that she can use the system without any guidance.

- Four participants agreed that the system was fun to use; unlike their general expectation of the use of a system for the first time. One participant commended the software; noting that much effort had been put to producing a good system. The participant however noted that there is still room for improvement.
- Five participants found it difficult to locate the resize button for the components. Three participants tried resizing the components from the borders. The general consensus was that they would have preferred resizing from any of the borders as it is done in the windows environment they are conversant with.
- All the participants found double clicking on the title bar to change the title somehow difficult. Four participants double clicked on the title bar icon instead. This suggests that an alternative method of changing the component title should be implemented.
- All the participants found it very easy to locate the basic window parts such as the buttons (minimise, restore, close) and the title bar.
- All the participants found minimising and restoring the component very easy. According to them, the icons were similar to what they are used to. This shows the importance of using familiar representations in the application. One participant however tried to restore a minimised component by dragging.
- Three participants complained that clicking the close button popped up a dialog asking them about deleting the component. This they found confusing as their intention was to close, not delete the component. Either the dialog box prompt or the functionality of that button should be changed to avoid confusing the users.
- It was easy for all the participants to find the icons representing the components. About 50% of the required components were located using the iconic representation, others were found using the tooltip. This shows the importance of using the right icons and the need to maintain the tooltips.
- Five participants were able to locate the edge icon based on the description given. One participant wasn't able to do this. When asked if they would have been able to locate the edge icon in the absence of the description given, they all answered negatively. This shows the need for some kind of help facility that identifies the component parts and provides brief descriptions of their use.
- One participant suggested that a tooltip help text be used to inform the user about what can be done to the edge links such as labelling them.
- Four participants frowned at the name "whiteboard" for the area where the components are displayed. They suggested the name be changed as the area in

question was not even white. One of the participants suggested the name “media workspace”.

- Adding components to the whiteboard which was expected to be a very simple task turned out to be a bit tricky. Four participants complained that the system did not make it obvious to the user that a component has been added to the whiteboard when the icon was clicked. General observations also showed that all the participants added a component more than once on at least one occasion. This shows the need to make it more obvious to the user that the selected component has been successfully added to the whiteboard. One of the participants also tried dragging the component icon from the toolbox to the whiteboard. This method of adding a component to the whiteboard is currently not supported but may be implemented as an alternative method.
- Two participants complained that the usual help facility was missing from the system. One of them even attempted bringing up the help feature by pressing the ‘F1’ functional key. Both users suggested that such facility be made available on the system.
- Two participants commented that the tooltips (help text that appear when the user places the mouse icon over an item briefly) are very useful. They suggested that the feature be maintained and more information added to make it more useful.
- One participant complained that ‘*right click*’ doesn’t bring up the usual menu he is familiar with such as ‘copy’, ‘paste’, etc. This is a limitation in Flash run time but should be looked into. A way of enabling the basic functionality users require when they right click should be implemented.
- The task sheet had an instruction that expected the user to add a video component. The participants complained about the fact that the icon in the toolbox was identified as ‘YouTube Video’ and not ‘video component’. This is an indicator that the video functionality has to be merged into one component that plays from different media sharing services (YouTube, Vimeo, etc).
- Three participants agreed that the system is (positively) memorable while one participant disagreed to this.
- One participant also commented that the toolbox seems to be cluttered with icons and as such another way of managing the icons should be investigated.

6.3.2 Multimedia Components Results

- One participant observed that the Twitter search component would be better in full screen mode to avoid scrolling horizontally and vertically. Providing a full screen mode is a trivial change and should be implemented.
- All the participants located the Twitter search component easily using the Twitter bird icon. However, most of them still paused to see the tooltip for confirmation. This shows the importance of using the right symbols to represent the components, and, unexpectedly, the value using the tooltip to supply confirmatory information rather than a helpful hint about the icon's functionality. This should be maintained, at least as is, if not augmented as above.
- One participant attempted to adjust the structure and path of the edge link by dragging the link. When asked why, the participant explained she would have loved to bend the link into a direction she prefers. This shows the need to implement edge routing for the edge links. This is not a trivial change to make and will therefore be left as future work.
- All the participants attempted to play the audio files immediately after they added the source but the file did not play immediately because it has to be selected on the playlist before the '*play*' button is clicked. This observation shows the need to ensure that the most recent audio source is pre-selected as soon as it is added to the playlist. This change is quite easy and has already been made.
- One participant suggested that a copyright warning be displayed when trying to upload an audio file so as to alert the user to the importance of uploading only audio files which they own.
- Only one participant found it difficult locating the PDP icon. The others located it quite easily.
- 50% of the participants preferred the first version of the PDP component; 20% preferred the second version while the rest 30% were indifferent. The most important reasons that informed their choice were step management and complexity. Figure 6.1 below show the important differences between both versions.

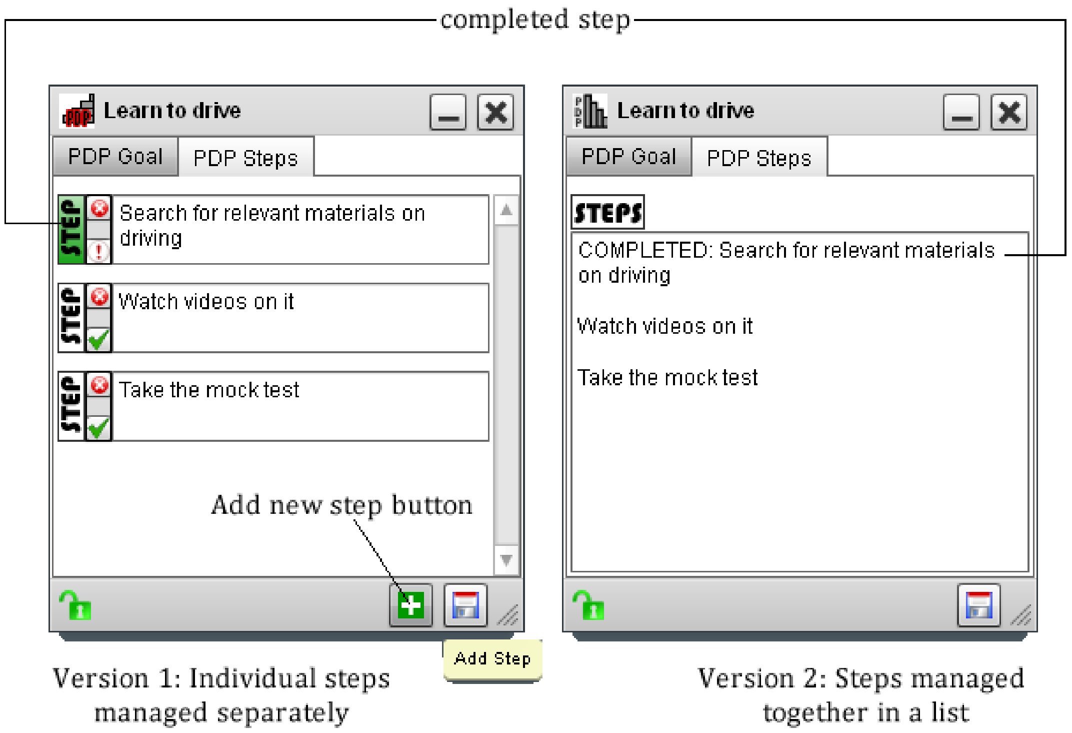

completed step
Learn to drive
PDP Goal
PDP Steps
STEP
Search for relevant materials on driving
Watch videos on it
Take the mock test
Add new step button
Add Step
Version 1: Individual steps managed separately
STEPS
COMPLETED: Search for relevant materials on driving
Watch videos on it
Take the mock test
Version 2: Steps managed together in a list

Figure 6.1 - Important differences between PDP versions

- Two participants agreed that the first version was too complex and not suitable for all users. One participant remarked that one has to be computer literate to be able to use the first version. The second version however was said to be self explanatory meaning that the user knows at once what is required. Two participants however frowned at having to number the steps by themselves in the second version demanding the system auto-number the steps when the ‘enter’ key is pressed.
- When asked which version of the component to improve and keep in the application, there were various answers from the participants. Two participants agreed that it would be nice to keep both versions. According to them, users can start from the easier one and then move on to the more complex one. One participant disagreed with this idea arguing that using components such as these is habit forming thus when a user is used to one version, the other one will be left unused. Another participant was indifferent as to which one to keep, and suggested that one be removed to avoid confusing the user.
- The general complaint for the first version was that it looked complicated and technical. One participant remarked that adding steps was difficult and a bit stressful while managing them later was easy and nice. Two other participants agreed that they liked the way steps are managed in the first version.

- Some of the participants suggested marking steps completed in the second version of the PDP component by simply writing the word ‘COMPLETE’ beside each step. This is a suggestion that might be useful. One way of relieving the user from typing the word (COMPLETE) all the time is to enable him select the step and click a button which then puts the word beside or in front of the selected step.
- One participant complained about the acronym PDP being used in the tooltip for the icon. The participant suggested that the full form should be used to avoid any confusion.
- Two participants remarked that the ‘Add step’ button was not easy to find on the first version of the component. Two other participants were observed to have clicked the ‘remove step’ button while trying to add a new step. The visual appearance of the steps management button on each step seems to confuse the user. A better visual arrangement of steps should be considered. My suggestion is the steps management buttons (delete step, mark as completed, mark as uncompleted) should be laid out on a ribbon at the bottom of the component. The steps should be made select-able with a visual cue to show when a step is selected. The icons at the bottom can then be used to manage the selected step.
- Two participants requested that the size of the PDP component be made bigger.
- Four participants complained that the help text in the steps field (*add action step...*) had to be deleted manually. They suggested that the help text should disappear when the user clicks to start typing. This is a necessary function and should have been implemented. An oversight must have led to it not being implemented and this has been immediately corrected.
- One participant suggested that undo should be possible when adding the steps.
- One participant suggested that links can be added directly to the steps rather than using the links component. This feature will be investigated to determine its usefulness.
- All the participants agreed that the PDP component is useful for planning learning. Three participants further remarked that apart from planning learning, the component can be used in other ways such as planning daily tasks, managing projects, preparing agenda for meetings, managing collaborative sessions such as group projects, etc. One participant, thinking of his forthcoming wedding, remarked that he would like to use the component as a wedding planner.

6.3.3 Simultaneous Evaluation Results

The multimedia learning spaces are multi-user spaces. This means that they are meant to be used simultaneously by many users at any given time. The system's ability to support this scenario was tested by conducting a simultaneous evaluation. The simultaneous evaluation session involved two postgraduate students from different disciplines using the application to create a personal development plan. The two participants used different computer systems to perform the required tasks. The tasks used for the session can be found in Appendix B. The evaluator was able to observe both participants simultaneously and a number of observations were noted.

The participants found the component easy to use and straight-forward. The difficulty they encountered however came from the locking mechanism of the component. The locking mechanism is used to control access to the component by multiple users. When a user is editing data, the other user cannot move the component, minimise close or resize it. The other user(s) can however edit data on the component simultaneously. Saving or updating the data is where the main challenge comes in. When one user updates the data being edited, the other user loses his data. This is because the update causes the model to send the 'data changed' notification which triggers the component to update its display interface with the latest data, thereby causing the current data to be lost. This challenge can be solved by saving the current data in a local variable before updating the component. After the update, the saved data can be restored to the component to ensure the user doesn't lose his current data. Another solution is providing a locking mechanism that is more restrictive.

Apart from the impediment described above, no other impediments were faced by the participants. Each participant performed their tasks successfully. Some tasks required one participant to interact with components added by the other participant. These tasks were performed successfully by both participants.

Interestingly, during the simultaneous evaluation session, one of the participants observed that when editing the edge link labels (not implemented as part of this project, but earlier by others), the label does not save changes on pressing the 'enter' key. This observation was noted and recorded immediately.

6.3.4 Summary of Evaluation Results and Recommendations

The cooperative evaluation session revealed some improvements to be made to the application. General improvements as well as specific improvements to components were suggested by the participants. Additional areas of improvement were also noted from observing the user interactions during the session and talking to them during the debriefing session. The following recommendations are therefore suggested for further improving the learning space application.

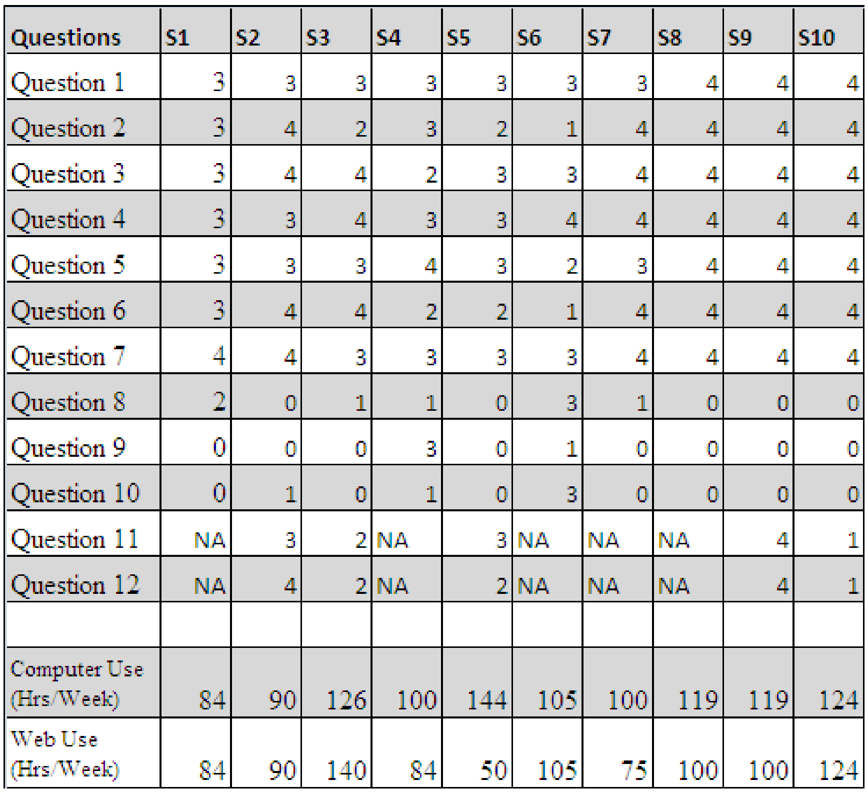
- A help facility should be provided for the application. If possible such help facility should be an interactive one; after all, the application is a multimedia application.
- An alternative name should be used instead of ‘whiteboard’; ‘media workspace’ has been suggested for this purpose.
- Resizing the components should be done using the borders of any of the four sides.
- An alternative method should be implemented for changing the component title
- The prompt to delete the component when the close icon is clicked should be changed to a more user friendly prompt.
- It should be possible to drag and drop icons on the whiteboard to add components in addition to the present method of adding components by clicking on an icon in the toolbox. Visual cues should also be used to inform the user that the component has been added successfully.
- If possible, right click should be modified to present ‘standard’ menu items such as copy and paste which were obviously ‘missed’ by the participants during the evaluation.
- Tooltip help texts should contain more explanatory texts, as the users depend on them more than was anticipated.
- A video component for streaming from multiple video sources should replace the existing one which streams from only YouTube.
- A copyright warning should be displayed when uploading any item from the user’s computer. If possible an input method should be used to get their confirmation of ownership before uploading begins.
- Edge routing should be implemented to enable the user draw links with corners if they are needed.
- The edge arrow should feature a tooltip text that explains what can be done with it when the mouse arrow is hovered above it. This should occur before the edge arrow is clicked.
- When the link label is being edited, it should save the changes when the user presses the ‘enter or return’ key on the keyboard.

- The first version of the PDP component should be simplified and retained while the second version should be removed to avoid confusing the user.

These points represented the main conclusions from the co-operative evaluations and the debrief sessions, whereas the next section discusses and analyses the feedback obtained from the questionnaires.

6.4 Questionnaire Results

Subjective evaluations regarding ease of use of the system and user satisfaction were collected via questionnaires. The questionnaires contained questions about the participant's knowledge of personal development planning; bipolar questions where the respondents rates his agreement or disagreement to a statement on a specified point scale and questions on how much the subject uses computer systems (including the internet). The questionnaire used can be seen in Appendix D. The result of the questionnaire is shown in the table below.



Questions	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
Question 1	3	3	3	3	3	3	3	4	4	4
Question 2	3	4	2	3	2	1	4	4	4	4
Question 3	3	4	4	2	3	3	4	4	4	4
Question 4	3	3	4	3	3	4	4	4	4	4
Question 5	3	3	3	4	3	2	3	4	4	4
Question 6	3	4	4	2	2	1	4	4	4	4
Question 7	4	4	3	3	3	3	4	4	4	4
Question 8	2	0	1	1	0	3	1	0	0	0
Question 9	0	0	0	3	0	1	0	0	0	0
Question 10	0	1	0	1	0	3	0	0	0	0
Question 11	NA	3	2	NA	3	NA	NA	NA	4	1
Question 12	NA	4	2	NA	2	NA	NA	NA	4	1
Computer Use (Hrs/Week)	84	90	126	100	144	105	100	119	119	124
Web Use (Hrs/Week)	84	90	140	84	50	105	75	100	100	124

Table 6.1 - Questionnaire Results

6.4.1 System Usability Scale

In order to determine the usability score for the system, a custom usability evaluation model was adapted from John Brooke's System Usability Scale (SUS) (Brooke, 1996). This scale is a simple ten-item scale that gives "a global view of subjective assessments of usability" (Brooke, 1996). The scale produces a score that represents the overall usability of the system being evaluated. The SUS was developed by Brooke at Digital Equipment Corporation (DEC) in 1986. It has been widely used in the evaluation of different systems ranging from online systems to mobile and desktop systems. In describing the results of 2,324 SUS surveys from 206 usability tests collected over a ten year period (Bangor *et al.*, 2008 cited in Bangor *et al.*, 2009), Bangor *et al.* established that the SUS was "highly reliable (alpha = 0.91) and useful over a wide range of interface types" (Bangor *et al.*, 2009). They also reported that gender had no effect on the SUS scores. However, there was a small, significant correlation between age and SUS scores (SUS scores decreasing with increasing age) (Bangor *et al.*, 2009). Tullis and Stetson (2004) carried out a comparison of questionnaires for assessing website usability with 123 participants. Among the questionnaires studied were

- System Usability Scale (SUS)
- Questionnaire for User Interface Satisfaction (QUIS)
- Computer System Usability Questionnaire (CSUQ)
- Words (adapted from Microsoft's Product Reaction Cards)
- A custom questionnaire of theirs

In concluding the study, they reported that "one of the simplest questionnaires studied, SUS (with only 10 rating scales), yielded among the most reliable results across sample sizes" (Tullis & Stetson, 2004). They also reported that the SUS was the only questionnaire whose questions address different aspects of the user's reaction to the system as a whole (Tullis & Stetson, 2004). The SUS has therefore been selected for scoring the usability of the system as described in the next sub section.

6.4.2 System Usability Score for the Questionnaires

In order to calculate this score, the sum of each question's contribution to the overall scale is first determined. In SUS, individual scores are not meaningful on their own (Brooke, 1996). Each item's score ranges from zero (0) to four (4). The individual score's contribution to the overall score is shown in the Table 6.2 below.

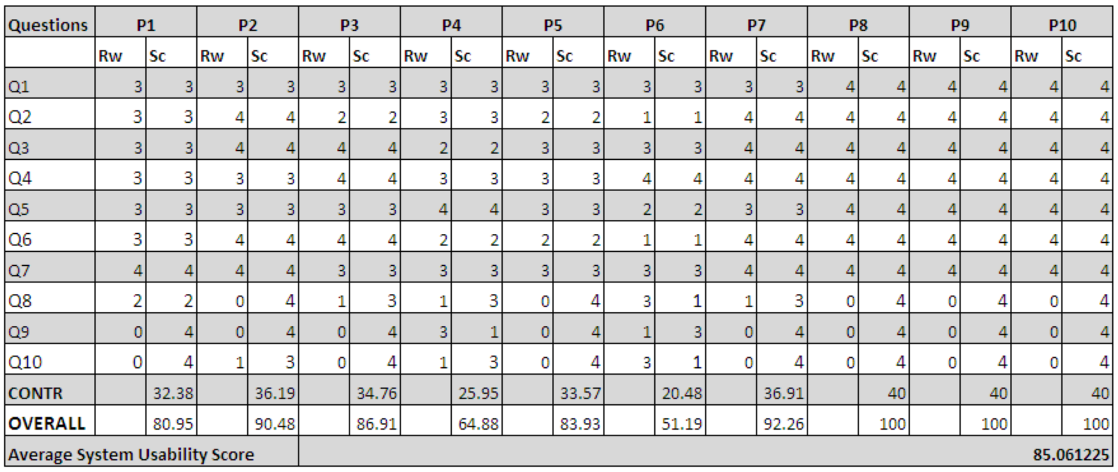
Question	Scoring Method	Max. Score
Q1	Scale Point	4
Q2	Scale Point	4
Q3	Scale Point	4
Q4	Scale Point	4
Q5	Scale Point	4
Q6	Scale Point	4
Q7	Scale Point	4
Q8	4 - Scale Point	4
Q9	4 - Scale Point	4
Q10	4 - Scale Point	4
Maximum Possible score		**40**

Table 6.2 - Scores contributed by individual questions on questionnaire

As shown in the table above, for items 1, 2, 3, 4, 5, 6, and 7, the score contribution is the exact scale position. Their contributions should have been scaled to minus one if the scale was between one and five instead (Brooke, 1996). For items 8, 9, and 10, the contribution is four (4) minus the scale position. To obtain the overall system usability score, the sum of the individual contributions is then multiplied by 2.5.

However, before multiplying the sum of the individual contributions by 2.5, some weighting has to be made accordingly to compensate for the customisations made to the evaluation questions. The positive questions (1, 2, 3, 4, 5, 6, and 7) and the negative ones (8, 9, and 10) will be scaled to have equal weight in the final score. In order to achieve this, the sum of individual scores for the positive questions will be multiplied by 0.7143 while that for the negative questions will be multiplied by 1.6667. These values were computed from the relative quantity of questions in each category and the total possible scores (20 for each category) for a balanced SUS system.

The table below shows the final contributions and overall score for each participant (A larger version of this table can also be found in Appendix D). The last row of the table shows the average usability score for the system.



Questions	P1		P2		P3		P4		P5		P6		P7		P8		P9		P10	
	Rw	Sc	Rw	Sc	Rw	Sc	Rw	Sc	Rw	Sc	Rw	Sc	Rw	Sc	Rw	Sc	Rw	Sc	Rw	Sc
Q1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	4	4	4	4	4	4
Q2	3	3	4	4	2	2	3	3	2	2	1	1	4	4	4	4	4	4	4	4
Q3	3	3	4	4	4	4	2	2	3	3	3	3	4	4	4	4	4	4	4	4
Q4	3	3	3	3	4	4	3	3	3	3	4	4	4	4	4	4	4	4	4	4
Q5	3	3	3	3	3	3	4	4	3	3	2	2	3	3	4	4	4	4	4	4
Q6	3	3	4	4	4	4	2	2	2	2	1	1	4	4	4	4	4	4	4	4
Q7	4	4	4	4	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4
Q8	2	2	0	4	1	3	1	3	0	4	3	1	1	3	0	4	0	4	0	4
Q9	0	4	0	4	0	4	3	1	0	4	1	3	0	4	0	4	0	4	0	4
Q10	0	4	1	3	0	4	1	3	0	4	3	1	0	4	0	4	0	4	0	4
CONTR		32.38		36.19		34.76		25.95		33.57		20.48		36.91		40		40		40
OVERALL		80.95		90.48		86.91		64.88		83.93		51.19		92.26		100		100		100
Average System Usability Score																				**85.061225**

Pn: Participant n **Rw**: Raw score obtained from participant **Sc**: Contributed score based on the scoring method (see Table 5 above)
CONTR: Sum of Contributions **OVERALL**: Overall usability score

Table 6.3 - Final contributions and overall SUS score

The graph below shows the usability score obtained from each participant's questionnaire.

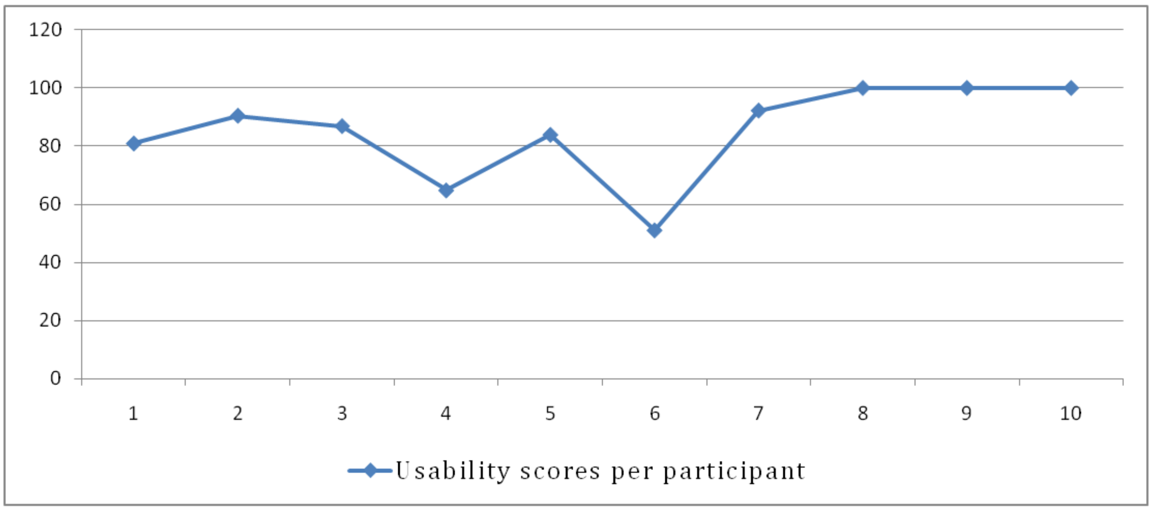


Figure 6.2 - Graph of usability scores per participant

The average usability score of the system is approximately eighty five percent (**85%**). It can be seen from the graph that only two points fell below the average usability score indicating a strong agreement by the participants on the usability of the system.

6.4.3 System Usability Result and Rate of Computer Usage

The graphs below show the relationship between the usability results and the rate of computer usage by the participants. Figure 6.5 shows the measure of correlation (linear dependence) between them also.

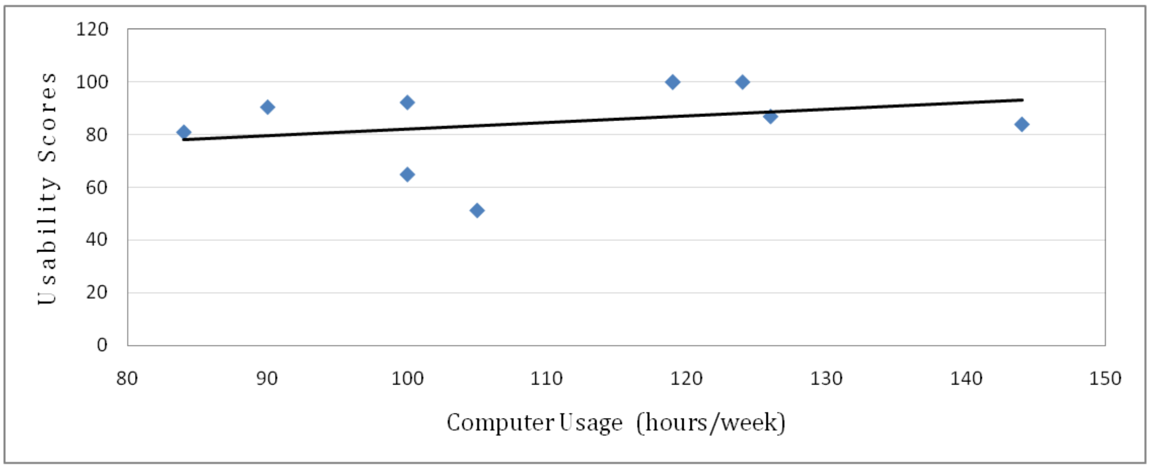


Figure 6.3 - Usability score against computer usage, with line of best fit

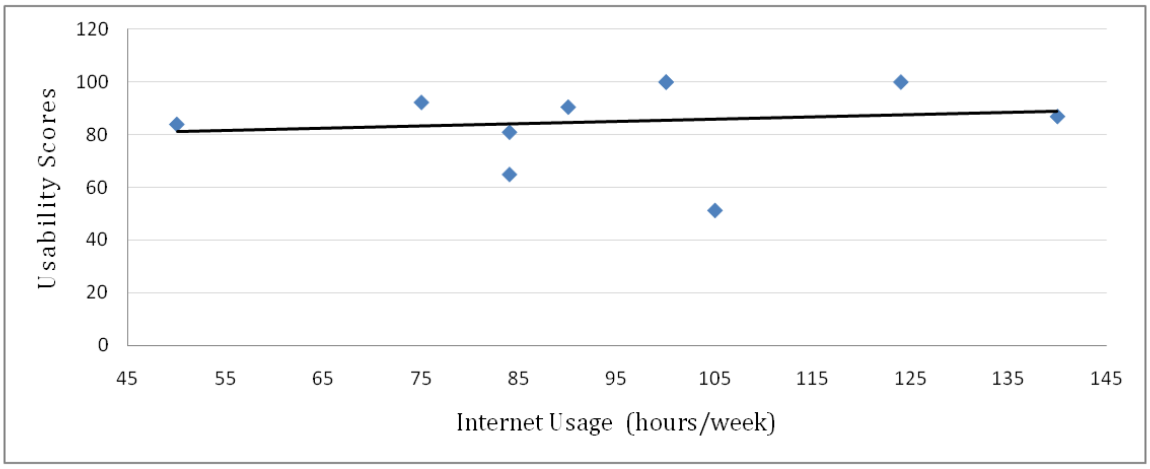


Figure 6.4 - Usability score against internet usage, with line of best fit

Correlation Coefficients			
	SUS	SUS	Interpretation
Computer Usage (hours/week)	**0.287446462**	**0.287446462**	Small Positive
Internet Usage (hours/week)	**0.130108853**	**0.130108853**	Small Positive
Statistical Function Used	Pearson (PPMCC)	Ms Excel Correl	N/A

Figure 6.5 - Correlation between computer usage and usability scores

The first plot (Figure 6.3) reveals a weak positive correlation between the hours spent using the computer or the internet (per week) and the individual usability scores. This is also confirmed by the correlation coefficients in Figure 6.5. This is an indication that the participants' familiarity with computers has little bearing on how usable they find the system.

The second plot (Figure 6.4) also shows a weak positive correlation (as confirmed by the coefficients again) between rate of internet use (hours/per week) and the system usability scores. This also indicates that the participants' familiarity with the internet has little bearing on how usable they find the system. The correlation in this case is weaker than for computer usage. This might be an indication that familiarity with desktop applications has more effect on how usable people find Rich Internet Applications (RIAs) than familiarity with traditional web applications. This needs to be investigated further as the sample size for this evaluation is not enough to draw a firm conclusion.

6.5 Research Questions

This section provides answers to the research questions which were raised at the beginning of the project in section 1.1. The questions and their respective answers are given below.

1. Can one use interactive multimedia spaces to actively support a learner in achieving his/her learning goals?
2. Can one develop a usable multimedia learning space that will run smoothly despite the limitations of the internet?
3. Can the usability of such multimedia learning space be evaluated by users?
4. Can a learner develop a personal development plan (PDP) in the multimedia learning space and use it to pursue a learning goal successfully?
5. Can a personal development plan so created by a learner be converted to a learning artefact which can be used and/or re-used by other learners to achieve similar outcome(s) as the initial creator/learner?
6. Can a learner share his personal development plans and pursue his learning goals collaboratively?
7. Can a learner recover components that have been deleted from the multimedia learning space intentionally or mistakenly?

The answer to the first question is yes. Interactive multimedia spaces can be used to actively support a learner. This is evident from the results of the user evaluation. According to the result of the evaluation, 60% of the participants strongly agreed that the system is useful for learning. Sharing knowledge is also a very good way of supporting a learner in achieving his goals according to the principle of constructionism (see section 2.4.2). With regards to sharing knowledge, 90% of the participants agreed that the system is useful for sharing knowledge. All the participants also agreed that the system is useful for planning personal development. These responses indicate that interactive multimedia learning spaces can be used to support a learner in achieving his/her learning goals.

The answer to the second question on whether a multimedia learning space can be developed to run smoothly despite internet limitations; appears to be yes. Although, the application was not evaluated over the internet, the research into the implementation of applications using Flex shows that this is possible. Flex and supporting technologies have been used widely to implement similar systems. Also, during the evaluation session (which was done on a local network of 6 computers), multiple '*rooms*' were instantiated on the Red5 back end server. The server handled all the connections efficiently and there was no report of a session crash. This is an indication that it can run smoothly on the internet which is just a wider network.

The answer to the third question (can the usability of such multimedia learning space be evaluated by users?) is yes; based on the results of the user evaluation. A customised usability scale based on the SUS was developed and used to successfully evaluate the system.

Can a personal development plan (PDP) be developed in the multimedia learning space and used to pursue a learning goal successfully? Based on the results of the evaluation, the answer to this fourth question is also yes. All the participants agreed that the system is useful for personal development planning. Most of the participants also agreed that the system is useful for learning and easy to use. Additionally, most of the participants commented at the end of the session that the system was fun to use. These feedbacks provide a very strong indication that when the improvements have been implemented, the system will be useful in personal development planning.

The answer to the fifth question on converting a personal development plan into a knowledge artefact is yes. After several refinements, the PDP component was successfully designed to support the user in converting his development plan into a knowledge artefact. This can be achieved by linking media components that relate to the development plan with the PDP component. Consequently, the richer the multimedia learning space application is in media components, the more the user will be able to generate and organise content relevant to his learning goals. Also, since the component is saved in the space, it can be reused by other learners. A better way to implement reuse by exporting the knowledge artefact is discussed in the future work section (see section 7.2).

The sixth question investigates if a learner can share his personal development plans and pursue his learning goals collaboratively. The answer to this is yes. First of all, from the results of the evaluation session, 90% of the participants agree that the application is useful for sharing knowledge. This raises the question of how willing they will be to do this, especially if it is not part of a course pedagogic practice. Secondly, the feedbacks from the simultaneous evaluation session showed that users can collaborative generate and utilise knowledge through the media components linked to the PDP component. The simultaneous evaluation session also indicated that in order to make this very effective, some improvement has to be made with regards to how the changes from each user is updated in the PDP component.

Can a learner recover components that have been deleted from the multimedia learning space intentionally or mistakenly? With regards to the current implementation of the system, the answer to this question is no. The available time frame for the project was not enough to implement this feature hence it has been suggested as future work with recommendations on how it can be achieved (see section 7.2).

6.6 Critical Analysis

This section critically examines the implementation of the multimedia learning spaces. It also highlights the strengths and weaknesses of the system from different perspectives.

6.6.1 Technical Analysis

The overall design of the multimedia system is technically sound. The MVCS architecture ensures that the system is loosely coupled on the client side and on the server side. The inversion of control strategy used further decouples the system by remove the weak link in encapsulation which is instantiation of objects. Furthermore, Flex architecture for building user interface components enables the developer to separate the view logic from the interface design and layout. The interface containers and controls are laid out in separate files (referred to as skin files) using MXML while the control and interaction logic for these user interface (UI) controls are written in ActionScript class files. This architecture ensures the system is well structured for future development and extension. The packages is also well structured, separating the components, events, services, etc into different packages which makes it easy to organise code within the application. There are some areas which could be improved. This includes the model persistence mechanism and the components' update mechanism.

The current method of persisting component data presently makes it a bit difficult to persist custom objects for components that require them. This is also a consequence of the ActionScript's method of passing object as references only. However, a careful study of the model layer can be done so that improvements can be implemented to allow persisting custom objects for components. This will save the developer the stress of having to create a component property for each item that needs to be persisted as it is currently. It will also ease the stress in separating complex components into a collection of smaller components that work together (e.g. the PDP Component). Such smaller components will be well encapsulated because data can be stored in custom objects for each component which can then be transferred through custom events for persistence.

The components' update mechanism is based on notifications from the model layer about changes to the model. In the current design, when multiple users are connected, changes made by one user must first be sent to the model and updated after which the components receive notification about the changes and then query the model for the update which is then used to update the display. In effect this implies that instances of the components are communicating with each other through the database on the server. This is not generally considered best practice because of the overhead of doing so. Additionally, this is not the intended purpose of database systems. Database systems are designed for medium to long term storage of data. This current update mechanism can be improved such that components can communicate directly with each other without passing through the database on the server. A new protocol from Adobe known as Real-Time Media Flow Protocol can be used to achieve this.

6.6.2 Real-Time Media Flow Protocol (RTMFP)

The Real-Time Media Flow Protocol (RTMFP) is "a new communication protocol from Adobe that enables direct end user to end user peering communication between multiple instances of the Adobe Flash Player client and applications built using the Adobe AIR framework" (Adobe, 2011e). RTMFP was release in 2008 and has been in use since then. RTMFP is different from RTMP is several ways among which are (Adobe, 2011e):

- RTMFP is based on the User Datagram Protocol (UDP), whereas RTMP is based on the Transmission Control Protocol (TCP). UDP based protocols are better than TCP based protocols in terms of efficiency of delivery when it comes to live streaming media. UDP based protocols also have decreased latency, increased audio quality and much greater connection reliability
- RTMFP supports sending data directly from one Adobe Flash Player Client to another unlike RTMP which must go through a server.

The diagrams below capture the main differences between RTMFP and RTMFP.

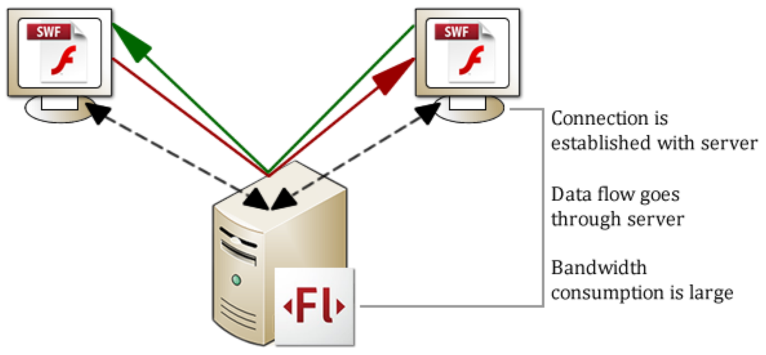

SWF
SWF
Connection is established with server
Data flow goes through server
Bandwidth consumption is large
Fl

Figure 6.6 - Communication using RTMP

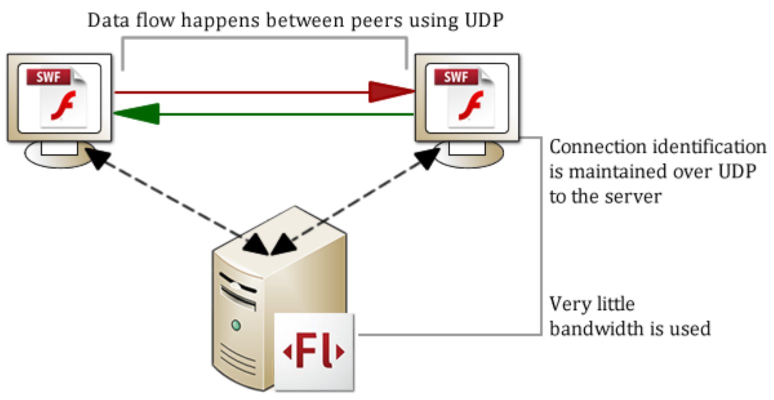

Data flow happens between peers using UDP
SWF
SWF
Connection identification is maintained over UDP to the server
Very little bandwidth is used
Fl

Figure 6.7 - Communication using RTMFP

Figure 6.6 shows the RTMP which requires a back end server to communicate with other client applications thereby routing data through the server. This also requires large bandwidth consumption. RTMFP on the other hand shown in Figure 6.7 enables clients to exchange data without going through the server. A connection will be established initially using the server but subsequent data exchange goes from one client to another while the connection identification is maintained on the server. RTMFP is also very secure. Unlike RTMP, RTMFP is "encrypted over the wire using an 128-bit AES encryption" (Hassoun & Heider, 2010). The receiving client is required to know the name of the stream and have the Peer ID (a 256-bit value associated with the publisher's identity) of the publisher to access the published stream.

RTMFP can be implemented in the multimedia learning spaces application such that the components can transmit data to each other for updates while the data required for persistence can be sent to the database server when necessary. This will prevent several round trips to the database server. RTMFP is supported by Flash Player 10 and above which is already supported by most browsers. The only limitation to implementing this is the lack of support for RTMFP by Red5 which is the current media server being used in the application. RTMFP is currently supported by Flash Media Server 4 only. As at the time of this writing, support for RTMFP is under development by the Red5 team so this is an area for future work.

6.6.3 User Interface Analysis

The user interface was extensively evaluated in the cooperative evaluation section (see section 6.3). The recommendations made by the participants and the subjective assessment gathered from the questionnaires will significantly improve the interface and user experience of the application when implemented, but retesting is advised as part of an iterative approach to user interface improvement (i.e. one set of changes may not make the user interface optimal). The results of the evaluation indicate that the user interface already provides significant benefits to the user; however some improvements can still be made in addition to those recommended in the evaluation results.

One area that can be improved upon is the ability to recognise when new components are added to the interface. Visual cues need to be implemented to ensure the user is aware that the new component has been added. One way of doing this might be to use a flashing border line around the component which disappears after a short time. Also, in terms of adding components, many users will attempt to double click the icons to add a component as observed during the evaluation sessions. This will result in adding two or more components to the whiteboard. A mechanism should be implemented to capture ‘double clicks’ and ensure only one instance is added to the whiteboard. Combining this with the visual cue already suggested will ensure the user enjoys a better experience with the application.

Another area of improvement in the user interface is the ability to track changes. There should be a way to notify the user who has returned to the space after leaving it for a while about changes that has occurred in the space. This is further discussed in the future work section (see section 7.2).

6.7 Summary

After successfully implementing the designs, the system was evaluated using cooperative evaluation with ten participants. This chapter has reported in detail how the evaluation was carried out, its results, their analysis and subsequent recommendations. The chapter began by examining the evaluation method used. It also discussed the choice of participants and the consideration of ethics to ensure the evaluation was performed in accordance with ethical requirements for postgraduate research. The results of the evaluation were reported in two ways; descriptively and analytically. Recommendations were then made to further improve the system. Finally a review of the research questions was performed and answers were provided for these research questions. The next chapter concludes the research by first summarising the achievements; identifying area for future work; reflecting on the approach and finally summarising the whole work.

Chapter 7

Reflection, Future Work and Conclusion

This section gives a summary of the achievements; areas of future work and reflections about the research methodology used and the project as a whole.

7.1 Summary of Achievements

This project has produced a number of significant achievements including:

- A clear description of the design of the existing spaces system which can be referred to in the future to understand the system for further development
- The addition of new multimedia components (PDP, Audio, Twitter Search and Space-Painter components) to the existing ones in the multimedia learning spaces application
- The production of a usable personal development planner, with support for collaborative generation of content for learning.
- Recommendations, mostly for improving the PDP component and some broader recommendations for improving the multimedia learning spaces application.
- A number of suggestions for future work in the existing application and also in the area of multimedia learning spaces applications in general.

7.2 Recommendations for Future Work

The following are recommendations for future work.

Exporting and Importing PDP Components for Reuse

After a user has successfully created a personal development plan and maybe converted it to a knowledge artefact, it should not be thrown away but preserved for reuse by other learners who wish to achieve similar learning goals. One way to preserve it is to leave it there in the learning space since the data is already stored in the database. While this is a good idea, it limits reuse of the artefact to the learning space alone implying that a learner who wishes to reuse the artefact must

- Use the same learning space application
- Be aware of the existence of the knowledge artefact.
- Know the location of the artefact (room or space id)

A better way to encourage reuse of knowledge artefacts created in the space is to implement a method and format for exporting them for reuse. Formats for export and data exchange can be created using eXtensible Markup Language (XML). The PDP data can be exported in custom XML tags; links, video and audio can also be exported with tags specifying their URLs; text components can be exported in XML CDATA tags, etc. The exported data can then be used in any application that can manipulate XML to extract and display the contents. A further advantage of exporting the PDP component and knowledge artefact is that the exported XML files can be listed on a page in the PLE software. This will increase discovery of the personal development plans through searching, tagging, rating, reviews, etc.

A way to import the exported data should also be implemented to further encourage reuse.

UNDO Support

One of the research questions was whether a component deleted from the media space (intentionally or otherwise) can be recovered. Time constraints prevented work in this area. However, the evaluation results show that this is a desired functionality. While there are many ways to implement undo support, the strategy chosen needs to be adapted to the current implementation of the application.

A good way to achieve undo within the application is to create a recycle bin component. Deleted components can be stored in the recycle bin component in chronological order from where they can be recovered when necessary. A question that immediately comes up here is 'how do we store the deleted component?' One way to store the component in the recycle bin is to serialise the object data and store it. However, the current implementation provides a better way to achieve this. Generally, when the application is executed, components are instantiated from data retrieved from the server. Thus all we need to store is the data for the deleted component which can be retrieved to re-create the components. This brings to mind another question; 'since there are different components with different data requirements, how do we know what data to store for each component type?' Fortunately, rather than implementing a complex 'if-then' structure to determine this; the current implementation also provides a solution. The '*BaseWhiteBoardView*' class is responsible for removing components through the '*removeComponentView*' method. In this method, a reference is available to the component being removed. A method can be implemented in the component such as '*getPersistedData'* which will return data that needs to be persisted for the component. This data can be attached to a custom event and dispatched to the recycle bin component for further processing.

Notification of changes since last visit

Space content may become large and complex. Finding new items inserted into a space by other users may therefore be hard to do. A '*find content I haven't seen*' function is needed to redress this.

One way of notifying users about changes since their last visit to the space should be implemented in the application. This can be done using visual cues such as an exclamation icon on the components whose data has changed since the user last visited the space. This function will require data to be stored about the last time the user visited the space. A history of 'important' changes in the components should also be stored and dated accordingly. Both data can be combined to determine which component(s) have changed since the last visit. A visual cue can then be displayed to notify the user about these changes. To avoid any ambiguity, it is important to note that leaving the application open for a while without interacting with it should also be taken into consideration. The last time the user interacted with the application should be considered as the 'last visit'.

Improved Update Mechanism

In section 6.6.2, a full description of the Adobe Real-Time Media Flow Protocol was discussed. This protocol can help remove the overhead incurred in handling component updates through the database server. Although RTMFP is only supported by Adobe Flash Media server presently, support for it is being built into the Red5 media server which is currently being used for the application.

7.3 Reflection and Approach Evaluation

This project has been relatively successful with over 85% of the research questions answered. Although earlier phases of the project were slow due to the time spent understanding the existing code base, the eventual result is good for the time available.

Particularly, the results of the user evaluation are encouraging. The approach of using more non-computer science students to evaluate the system produced a rounded set of recommendations from the general user population for the future PLE. The results of the evaluation will be very useful for future development.

The research methodology is worth reviewing: The project started with a “Preliminary Preparation” phase during which background research was carried out including the survey of relevant literature. This phase was immediately followed by the “Design, Development and Testing” phase which involved understanding the existing code base, modifying it and then extending it by implementing the new designs. The “Report, Review and conclusion” phase then followed which produced this dissertation. It is felt that this structure was apt for the project and produced good results.

Methodologically, improvements can be made to a project process like this in the future to improve success rates in projects like this; for example, the creation of more milestones in each phase. Milestones within a project help the enacting student on track. When a task is due for submission (whether officially or otherwise), more effort is put in to ensure its completion. Thus having more milestones can improve the chances of successfully completing similar projects in time.

7.4 Conclusion

This dissertation has presented the outcome of the research project on usability in user generated learning spaces. The goal of the project was to investigate usability in user generated learning spaces in order to improve the usability of an existing multimedia learning space application in the Manchester Personal Learning Environment (PLE).

The project involved the extension of the learning space application to effectively support learning by improving the user interface and also adding more components to the application. The research involved technical aspects as well as user interface and user interaction design.

The introductory chapter presented an overview of learning spaces and the motivation for the research. Research question were raised that needed to be answered at the end of the project. The project goal was set and also split into objectives to ensure it is achieved.

The background chapter explored the project topic within a wider research context. Important terms including learning, theory, cognition, etc were defined. Different theories of learning and their application in learning were discussed. User generated content; social learning environment and constructionism were also discussed showing the link between these concepts and personal learning or development. A review of the literature on Personal Development Planning (PDP) was carried out and it was shown that PDP is a good way to support learning.

Supporting learning requires understanding of how learning happens and also, where it does. Background research was conducted investigating the trends in designing physical and virtual learning spaces. Since virtual learning spaces are designed to support learning through technology based tools, it is important to assess the usability of such virtual learning spaces to ensure they meet their technical as well as pedagogic goals. This prompted the research on usability of learning spaces. The concept of usability was well explored in relation to system acceptability. The background chapter was concluded with a description of the architecture of an existing Personal Learning Environment (PLE) that supports virtual learning spaces.

The system design chapter detailed the design of the system generally. All implicit assumptions in the requirement specifications were identified and analysis conducted to determine the basic building blocks for the system as well as set the design goals of the system. The essential components that make up the existing multimedia learning space application were identified including the media server; the space client application and the RTMP connector. The layout and architecture of the system were described covering the user interface as well as the structure of the entire system. The MVCS architecture used was complemented with the inversion of control approach to further ensure the system is decoupled and easy to extend in the future.

The component design chapter covered the design of components to be added to the existing ones in the application. Each component was carefully designed to meet the design specifications. The designs were evaluated and refined until acceptable ones were produced.

The implementation chapter detailed how the designs were implemented. The tools chosen were discussed and reasons given for their choice. Challenges encountered during this phase were also presented and explanations given as to how these challenges were solved.

The evaluation and analysis chapter described the user evaluation of the implemented software. The type of evaluation; tasks for the evaluation and choice of human participants were discussed. The user evaluation was required to conform to standard research ethics which was also reported accordingly. The results of the evaluation were presented in descriptive as well as analytic manner. A summary of the results and recommendations for improvements were also given. Additionally, answers were presented for the research questions raised at the beginning of the project. Finally, a critical analysis was conducted on the software and reported also.

The concluding chapter gave a summary of the achievements made during the project. Possible areas for future work were identified and detailed also. After these, a reflection was done on the project and the research methodology. The approach taken was evaluated and suggestions were made for future improvement in the approach to ensure it is more effective in result delivery.

In conclusion, this project was focused on designing spaces to support learning. Physical spaces as well as virtual spaces can be designed to support learning. However, both are made from different raw materials even though they seek to achieve similar goals. The architect designs physical learning spaces from ‘bricks and mortars’ using the landscape as his canvas while the technologist develops virtual learning spaces from ‘bits and bytes’ using electronic devices as his canvas. The learning spaces so designed become the learner’s canvas upon which he constructs his learning. It is therefore very important to design usable learning spaces that transcend the awe of architectural magnificence and the illusions of interactive widgets to achieve the aim of the learner which is very simple: ‘to learn’.

The “Net Generation Learners” (Oblinger & Oblinger, 2005) are evolving learning in very unpredictable ways. When creating learning spaces for this breed of learners, “you can’t be sure how these spaces will be used. You are just creating the opportunities for things to happen” (Tom Finnigan cited in JISC, 2006). Trends in Information and Communication Technology will continuously redefine the meaning, boundaries and styles of learning. The challenge however is that while technology comes and goes, the psychology of how people learn is more persistent (Brown & Long, 2006). Therefore, we must return to the pedagogical roots (learning theories and styles) in order to be able to blend technology, learning goals and today's learners' in the right learning spaces.

Oblinger (2006b) recommends involving the users in learning space design. Based on the observations and results of the user evaluation, I agree with her. This includes "students, faculty, and staff". This is a vital point because while architects and developers see the complexities involved in realising a design, learners do not. What they see is the kind of learning they wish to have in the environment of their choice. Thus it becomes a challenge to the architect and developer to bring to reality the wishes of the learners. This is the reason why participatory design and/or user evaluation and co-design as practiced in this project are the only realistic approaches to the construction of usable learner spaces.

List of References

Adobe Systems Incorporated, 2009. *The RTMP Specification 1.0* [online]. Available at: http://wwwimages.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp_specification_1.0.pdf. [Accessed 12 June, 2011].

Adobe Systems Incorporated, 2009c. *Flash Media Server 3.5 - Core server architecture*. [online]. Available at: http://help.adobe.com/en_US/FlashMediaServer/3.5_TechOverview/WS5b3ccc516d4fbf351e63e3d119ed944a1a-7ffa.html. [Accessed 30 July, 2011].

Adobe Systems Incorporated, 2011. *What is Flex*? [online]. Available at: http://www.adobe.com/products/flex/. [Accessed 25 April, 2011].

Adobe Systems Incorporated, 2011b. *Adobe Flash Media Server Family*. [online]. Available at: http://www.adobe.com/products/flashmediaserver/. [Accessed 22 July, 2011].

Adobe Systems Incorporated, 2011c. *Real-Time Messaging Protocol (RTMP) specification*. [online]. Available at: http://www.adobe.com/devnet/rtmp.html. [Accessed 7 July, 2011].

Adobe Systems Incorporated, 2011d. *Flex framework*. [online]. Available at: http://www.adobe.com/products/flex/flex_framework/. [Accessed 17 July, 2011].

Adobe Systems Incorporated, 2011e. *Flash Media Server 4; FAQ for Real-Time Media Flow*. [online]. Available at: http://www.adobe.com/products/flashmediaserver/rtmfp_faq/. [Accessed 20 August, 2011].

ASDocs4 (Adobe Systems Incorporated), 2011. *ActionScript Documentation for Adobe Flex 4*. [online]. Available at: http://help.adobe.com/en_US/Flex/4.0/UsingSDK/index.html. [Accessed 2 June, 2011].

AskMeFlash, 2009. *Comparison Wowza vs FMS vs Red5*. [online]. Available at: http://askmeflash.com/article/10/comparison-wowza-vs-fms-vs-red5. [Accessed 30 July, 2011].

Bangor, A., Kortum, P., and Miller, J., 2009. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies*, 4(3), pp.114-123.

Bergan, S., 2007. *Qualifications: introduction to a concept*. Strasbourg: Council of Europe Publishing

Berkovitz, J., 2006. *Flex Best Practices: Applying Design Patterns and Architecture*. [online]. Available at: http://joeberkovitz.com/max2006/RI304W_FlexBestPractices_JoeBerkovitz.ppt. [Accessed 12 July, 2011].

Brooke, J., 1996. SUS: a “quick and dirty” usability scale. **In**: P. W. Jordan, B. Thomas, B. A. Weerdmeester, & A. L. McClelland. *Usability Evaluation in Industry*. London: Taylor and Francis.

(Also see http://hell.meiert.org/core/pdf/sus.pdf)

Brown M., 2005. "Learning Spaces". In: D. Oblinger and J. Oblinger, eds. *Educating the Net Generation*. Boulder, Colo.: EDUCAUSE, 2005. Ch. 12.

Brown, M. and Long, P., 2006. "Trends in Learning Space Design". In: D. Oblinger, ed. 2006. *Learning Spaces.* Boulder, Colo.: EDUCAUSE. Ch. 9.

Chism N. V. N., 2006. "Challenging Traditional Assumptions and Rethinking Learning Spaces". In: D. Oblinger, ed. 2006. *Learning Spaces.* Boulder, Colo.: EDUCAUSE. Ch.2.

Chowdhury, M. S., 2006. Human Behavior In The Context of Training: An Overview Of The Role of Learning Theories as Applied to Training and Development. *Journal of Knowledge Management Practice*, 7(2).

Clever, N., Kirchner, A., Schray, D. & Schulte, M., 2009. *User Generated Content.* [Essay], Institut für Wirtschaftsinformatik. Westfälische Wilhelms-universität. Münster.

Cottrell, S., 2010. *Skills for success: the personal development planning handbook (second edition)*. Basingstoke: Palgrave Macmillan.

Crawford, K., 1996. Vygotskian approaches to human development in the information era. *Educational Studies in Mathematics*, (31), pp. 43-62.

Dechant, E. V., 1991. *Understanding and teaching reading: an interactive model*. Hillsdale, N.J., L. Erlbaum Associates.

DeVries, R., 2002. *Developing Constructivist Early Childhood Curriculum: Practical Principles and Activities*. New York, Teachers College Press.

Dillenbourg, P., Schneider, D., and Synteta, V., 2002. “Virtual Learning Environments”. *Proceedings of the 3rd Congress on Information and Communication Technologies in Education*, Rhodes, Kastaniotis Editions, Greece, pp.3-18.

Domjan, M., and Burkhard, B., 1993. *Domjan and Burkhard's The principles of learning and behavior*. Pacific Grove, Calif, Brooks/Cole Pub. Co.

Doukidis, G. I., Mylonopoulos, N., & Pouloudi, N., 2004. *Social and economic transformation in the digital era*. Hershey: Idea Group Pub

Ethics-CS-UoM, 2010, *Welcome to Computer Science Ethics - Keep Calm*. [online]. Available at: http://ethics.cs.manchester.ac.uk/. [Accessed 17 August, 2011].

Eustace, E., 2009. *A survey of Inversion of Control frameworks for Flex*. [online]. Adobe Flex Developer Center. Available at: http://www.adobe.com/devnet/flex/articles/ioc_frameworks.html. [Accessed 20 July, 2011].

Fortin, C., & Rousseau, R. (1989). *Psychologie cognitive: une approche de traitement de l'information* - - .

Fritscher, L., 2011. *Cognitive Theory - Definition of Cognitive Theory* (Reviewed by The Medical Review Board). [online]Available at: http://phobias.about.com/od/glossary/g/cognitivethedef.htm [Accessed 16 March 2010].

Fruchter, R. and Emery, K., 1999. Teamwork: assessing cross-disciplinary learning. In: C. M. Hoadley and J. Roschelle, eds. *Proceedings of the 1999 conference on Computer Support for Collaborative Learning (CSCL '99), International Society of the Learning Sciences*, Article 19.

Gamma, E., Johnson, R., Vlissides, J., Helm R., 1995. *Design patterns: elements of reusable object-oriented software*. Reading, Mass, Addison-Wesley.

Gosling, D., 2003. *Personal development planning*. Birmingham: Staff and Educational Development Association.

Graetz K. A., 2006. "The Psychology of Learning Environments". In: D. Oblinger, ed. 2006. *Learning Spaces*. Boulder, Colo.: EDUCAUSE. Ch. 6.

Haase, C., 2010. *Flex 4 fun*. Mountain View, Calif: Artima.

Hart, J., 2009. Building a social learning environment - for free or at low cost Part 1: Using free, public social media tools. *Inside Learning Technologies Magazine (October 2009)*.

Hart, J., 2011, *Top 100 Tools for Learning 2011*. [online]. Available at: http://c4lpt.co.uk/top-tools/top-100-tools-for-learning-2011/. [Accessed 20 August, 2011].

Hassoun, D., & Heider, J., 2010. *Peer-assisted networking using RTMFP groups in Flash Player 10.1*. [online]. Available at: http://www.adobe.com/devnet/flashmediaserver/articles/p2p_rtmfp_groups.html. [Accessed 20 August, 2011].

Hertzum, M., and Jacobsen, N. E., 2001. The Evaluator Effect: A Chilling Fact about Usability Evaluation Methods. *International Journal of Human-Computer Interaction*, 13(4), pp. 421-443.

Illeris, K., 2007. *How we learn: learning and non-learning in school and beyond*. London, Routledge.

International Organization for Standardization, 2008. *ISO 9241-151: Ergonomics of human-system interaction - Part 151: Guidance on World Wide Web user interfaces*. Geneva: ISO.

Jackson, N., 2001. *Personal Development Planning: What does it mean? PDP Working Paper 1, Version 4*. Learning and Teaching Support Network (LTSN) Generic Centre. Available at: http://www.heacademy.ac.uk/resources/detail/resource_database/id65_Personal_Development_Planning_What_does_it_mean. [Accessed 01 August 2011].

Joint Information Systems Committee (JISC), 2006. *Designing Space for Effective Learning: A Guide to 21st Century Learning Space Design* (Bristol, U.K.: Higher Education Funding Council for England, 2006), Available at: http://www.jisc.ac.uk/uploaded_documents/JISClearningspaces.pdf [Accessed 15 March, 2011]

Jonassen, D. H., and Land, S. M., 2000. *Theoretical foundations of learning environments*. Mahwah, N.J., L. Erlbaum Associates.

Kadle, A., 2010. *Elements For Constructing Social Learning Environments*. [online]. Available at: http://www.upsidelearning.com/blog/index.php/2010/03/10/elements-for-constructing-social-learning-environments/. [Accessed 8 August, 2011].

Kim, T. and Axelrod, S., 2005. Direct Instruction: An Educators' Guide and a Plea for Action. *The Behavior Analyst Today*, 6(2), p. 111.

Krumm, J., Davies, N., Narayanaswami, C., 2008, User Generated Content. *IEEE Pervasive computing*, 7 (4), pp.10-11.

Languages, Linguistics and Cultures, University of Manchester (LLC-UoM), 2010. Personal Development Plan (PDP). Available at: http://www.llc.manchester.ac.uk/intranet/ug/pdp/. [Accessed 15th July, 2011].

Larman, C., 2005. *Applying UML and patterns: an introduction to object-oriented analysis and design and iterative development*. Upper Saddle River, N.J., Prentice Hall PTR.

Learning Theories Knowledgebase (LTKB), 2011. *Behaviorism*. [online] Avaliable at: http://www.learning-theories.com/behaviorism.html [Accessed 14 April 2011].

Learning Theories Knowledgebase (LTKB), 2011a. *Social Development Theory (Vygotsky)*. [online] Avaliable at: http://www.learning-theories.com/vygotskys-social-learning-theory.html [Accessed 14 April 2011].

Leonard, K., Noh, E.K., and Orey, M., 2007. "Introduction to Emerging Perspectives on Learning, Teaching, and Technology". In: M. Orey, ed. *Emerging Perspectives on Learning, Teaching, and Technology*. [online] Available at: http://projects.coe.uga.edu/epltt/ [Accessed 15 March 2011].

Lindgaard, G., Brown, A., & Bronsther, A., 2005. Interface Design Challenges in Virtual Space. **In**: D. R. F. TAYLOR, ed. 2005. *Cybercartography: theory and practice (Modern Cartography Series)*. Amsterdam: Elsevier Science. pp. 211-229

Marx, M. H., 1970. *Learning: Theories*. New York, Macmillan Co.

Monk, A., Davenport, L., Haber, J., & Wright, P., 1993. *Improving your human-computer interface: a practical technique*. New York, Prentice Hall.

NCIHE (National Committee of Inquiry into Higher Education), 1997. *Higher Education in the Learning Society, (the Dearing Report)*. London: HMSO.

Neill, S., and Etheridge, R., 2008. Flexible Learning Spaces: The Integration of Pedagogy, Physical Design, and Instructional Technology. *Marketing Education Review,* 18(1), pp.47-53., Available via Business Source Premier, EBSCOhost, [Accessed 17 April 2011].

Nielsen, J., 1993. *Usability engineering*. New York, Academic Press, Inc.

Nokelainen, P., 2006. An empirical assessment of pedagogical usability criteria for digital learning material with elementary school students. *Journal of Educational Technology & Society*, 9(2), pp.178–197.

Norman, D. A., 2000. *The Design of Everyday Things*. London, The MIT Press.

Oblinger D., 2006a. "Space as a Change Agent". In: D. Oblinger, ed. 2006. *Learning Spaces*. Boulder, Colo.: EDUCAUSE. Ch. 1.

Oblinger D., 2006b. "Learning How to See". In: D. Oblinger, ed. 2006. *Learning Spaces.* Boulder, Colo.: EDUCAUSE. Ch. 14.

Oblinger D., and Oblinger J., 2005. "Introduction". In: D. Oblinger and J. Oblinger, eds. *Educating the Net Generation*. Boulder, Colo.: EDUCAUSE, 2005. Ch. 1.

O'Connell, C., 2001. Case Study F - The University of Manchester, Moving Personal Development Planning from the Periphery to the Mainstream. **In**: *Personal development planning: institutional case studies: University of Wales Bangor et al*. York: LTSN.

Orlando, D., 2009. *Second Generation Micro-architectures, Part 1: The Swiz Framework*. [online]. Available at: http://danorlando.com/?tag=swiz-framework. [Accessed 12 July, 2011].

Papert, S., 1986. *Constructionism: A New Opportunity for Elementary Science Education*. Massachussetts Institute of Technology, Media Laboratory, Epistemology and Learning Group.

Papert, S, & Harel, I., 1991. *Constructionism: research reports and essays, 1985-1990*. Norwood, N.J., Ablex Pub. Corp.

QAA (Quality Assurance Agency for Higher Education), 2001. *Guidelines for HE Progress Files*. [online]. Available at: http://www.qaa.ac.uk/Publications/InformationAndGuidance/Documents/progfile2001.pdf. [Accessed 25 July, 2011].

Quinton, S., & Smallbone, T., 2008. PDP implementation at English universities: what are the issues? *Journal of Further and Higher Education*, 32, pp.99-109.

Red5 Team, 2010. *Red5 Teams Releases 1.0 RC Build*. [online]. Available at: http://www.red5.org/2010/12/31/release_1_0_build/. [Accessed 30 July, 2011].

Research Methods Knowledge Base (RMKB), 2006. *Introduction to Evaluation*. [online]. Available at: http://www.socialresearchmethods.net/kb/intreval.htm [Accessed 04 May, 2011]

Richardson, V., 2003. Constructivist Pedagogy. *Teachers College Record*, 105(9), pp.1623 – 1640.

Rughani, A., Franklin, C., & Dixon, S., 2003. *Personal development plans for dentists: the new approach to continuing professional development*. Abingdon, Oxon, Radcliffe Medical Press.

Shneiderman, B., 1980. *Software psychology: human factors in computer and information systems*. Cambridge, Mass, Winthrop Publishers.

Standridge, M., 2002. “Behaviorism”. In: M. Orey, ed. *Emerging Perspectives on Learning, Teaching, and Technology*. [online] Available at: http://projects.coe.uga.edu/epltt/ [Accessed 15 March 2011].

Sudgen, T., 2009. *The Trend Towards Inversion-of-Control Frameworks in Flex*. [online]. Adobe Consulting. Available at: http://blogs.adobe.com/tomsugden/2009/07/the_trend_towards_inversionofc.html. [Accessed 10 August, 2011].

Swiz Team, 2011. *[S] Swiz Framework: The brutally simple micro-architecture for Enterprise ActionScript development*. [online]. Available at: http://swizframework.org/. [Accessed 12 June, 2011].

THEA (The Higher Education Academy), 2011. *PDP - Personal Development Planning*. [online]. Available at: http://www.heacademy.ac.uk/resources/detail/pdp/pdp. [Accessed 25 July, 2011].

Tullis, T.S., & Stetson, J.N., 2004. A Comparison of Questionnaires for Assessing Website Usability. *Usability Professional Association Conference Proceedings, 2004*.

Ughade A., Raffin E., Bouteiller G., Bell I., Cocon P., 2007.*Personal Learning Environment: Design Factors and Implementation*. Msc. The University of Manchester.

van Harmelen, M., 2010. *Short introduction: the Manchester PLE*. [online]. Available at: http://www.youtube.com/watch?v=Ooy2xR9YcBk. [Accessed 04 May, 2011]

van Harmelen, M., 2011. *Discussion on Usability in User Generated Learning Spaces*. [email] (Personal communication, 27 April 2011).

van Harmelen, M., 2011b. *Discussion on the design of the PDP Component*. [conversation] (Personal communication, 21 July 2011).

Vygotsky, L. S., and Cole, M., 1978. *Mind in society the development of higher psychological processes*. Cambridge, Mass. [u.a.], Harvard Univ. Press

Ward, R., 2001. *Personal development planning: institutional case studies: University of Wales Bangor et al*. York: LTSN.

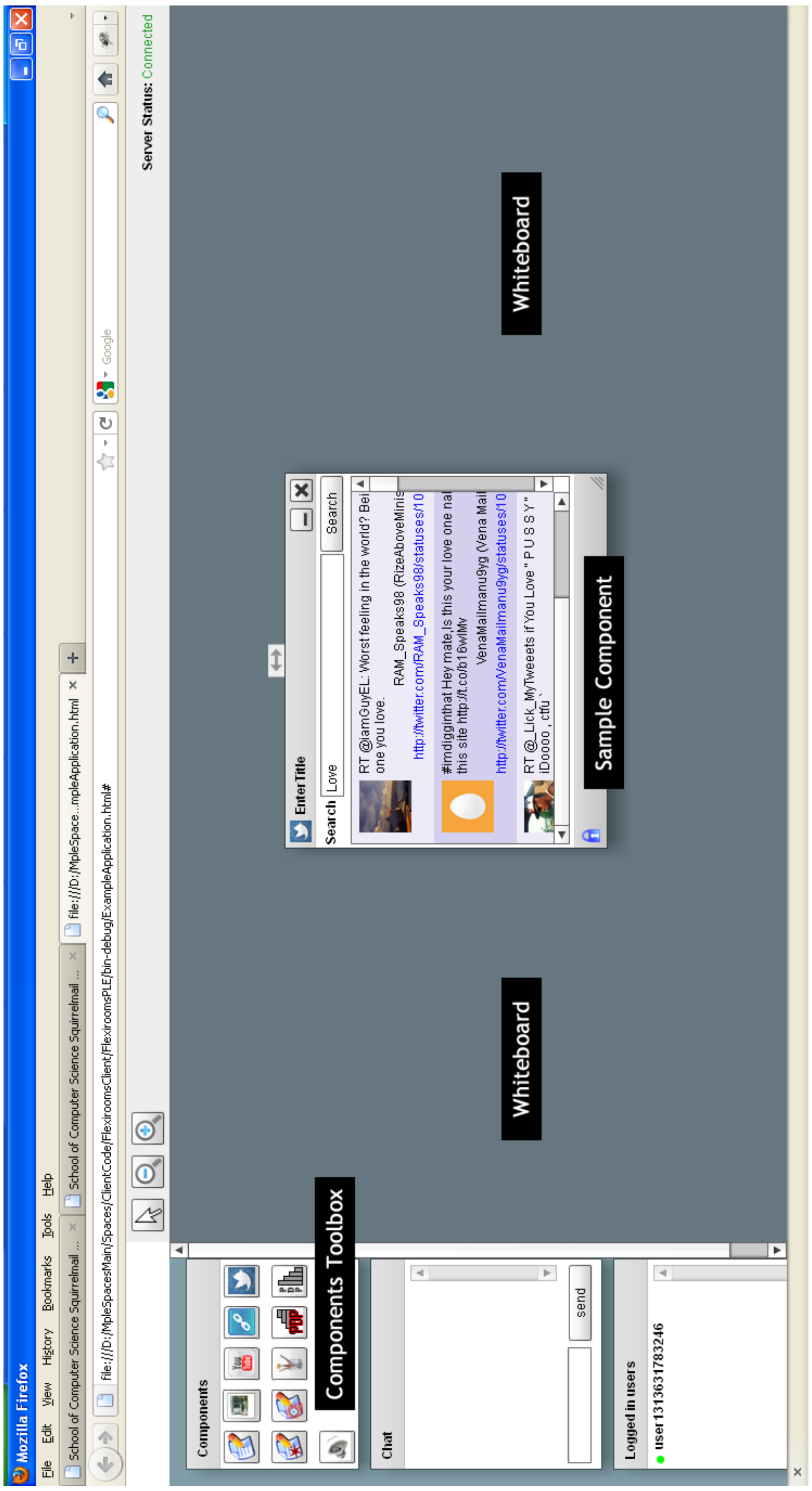
Wortham, S., 2003. Learning in Education. *Encyclopedia of Cognitive Science*, Volume 1, Article 563, pp.1079-1082.

Wunsch-Vincent, S., & Vickery, G., 2007. *Participative Web and user-created content: Web 2.0, wikis and social networking*. Paris: Organisation for Economic Co-operation and Development.

Zaharias, P, and Poylymenakou, A., 2009. Developing a Usability Evaluation Method for e-Learning Applications: Beyond Functional Usability. *International Journal of Human-Computer Interaction*, 25(1), pp. 75-98. Available via Business Source Premier, EBSCOhost, [Accessed 11 April 2011].

Appendix A

Application Screenshots


Mozilla Firefox
File Edit View History Bookmarks Tools Help
School of Computer Science Squirrelmail ...
School of Computer Science Squirrelmail ...
file:///D:/MpleSpace...mpleApplication.html
file:///D:/MpleSpacesMain/Spaces/ClientCode/FlexiroomsClient/FlexiroomsPLE/bin-debug/ExampleApplication.html#
Google
Server Status: Connected
Components
Components Toolbox
Chat
send
Logged in users
user1313631783246
Whiteboard
Whiteboard
EnterTitle
Search
Love
Search
Sample Component

RT @iamGuyEL: Worst feeling in the world? Bei
one you love.
RAM_Speaks98 (RizeAboveMinis
http://twitter.com/RAM_Speaks98/statuses/10

#imdigginthat Hey mate,Is this your love one nal
this site http://t.co/b16wlMv
VenaMailmanu9yg (Vena Mail
http://twitter.com/VenaMailmanu9yg/statuses/10

RT @_Lick_MyTweeets if You Love " P U S S Y "
iDoooo , ctfu `

Figure A.1 – Screenshot of the application layout

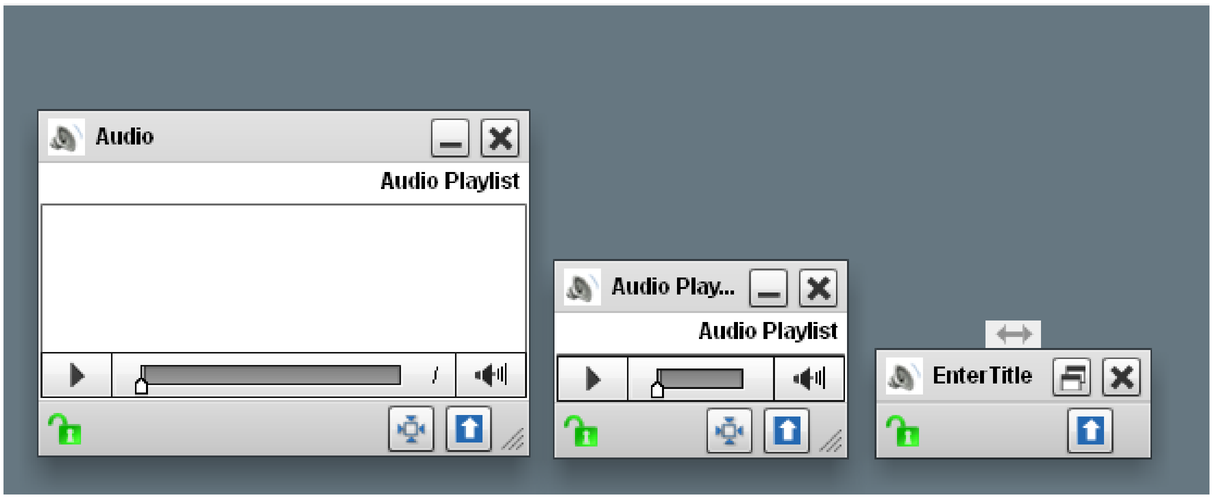
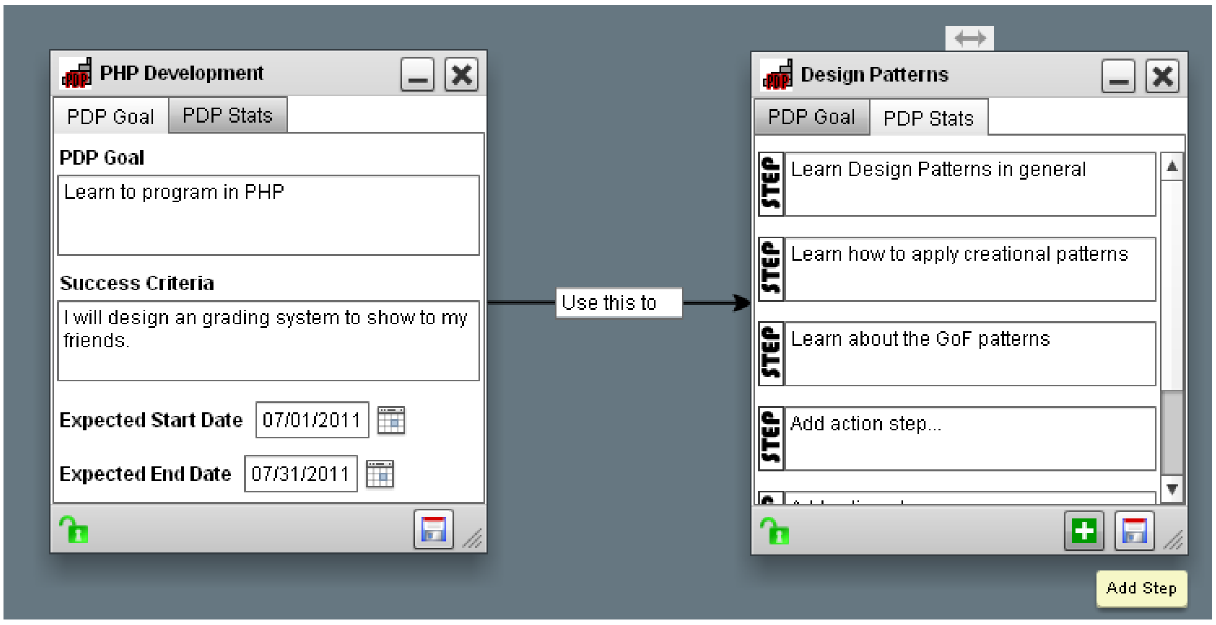


Figure A.2 – The Audio Component in different states (Normal, Resized, Minimised)



PDP Goal

Learn to program in PHP

Success Criteria

I will design an grading system to show to my friends.

Expected Start Date 07/01/2011

Expected End Date 07/31/2011

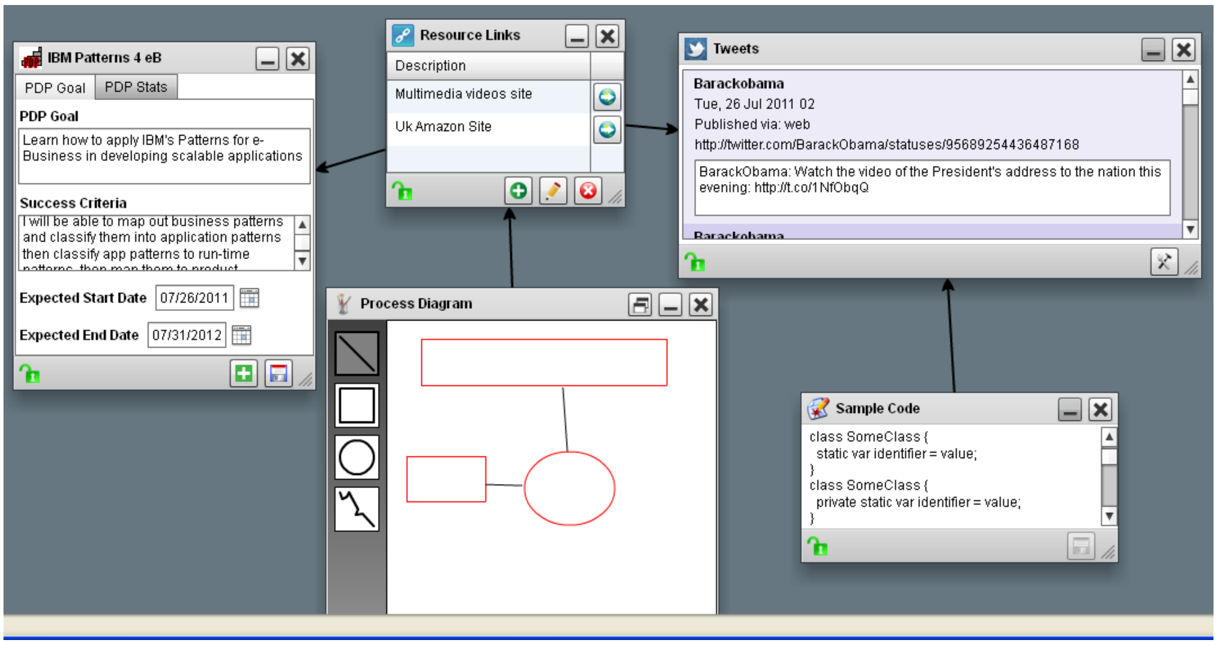
Learn Design Patterns in general

Learn how to apply creational patterns

Learn about the GoF patterns

Add action step...

Figure A.3 – Creating a Personal Development Plan using the PDP Component



IBM Patterns 4 eB

PDP Goal | PDP Stats

PDP Goal

Learn how to apply IBM's Patterns for e-Business in developing scalable applications

Success Criteria

I will be able to map out business patterns and classify them into application patterns then classify app patterns to run-time patterns, then map them to product

Expected Start Date 07/26/2011

Expected End Date 07/31/2012

Resource Links

Description
Multimedia videos site
Uk Amazon Site

Tweets

Barackobama
Tue, 26 Jul 2011 02
Published via: web
http://twitter.com/BarackObama/statuses/95689254436487168

BarackObama: Watch the video of the President's address to the nation this evening: http://t.co/1NfObqQ

Barackobama

Process Diagram

Sample Code

```
class SomeClass {
  static var identifier = value;
}
class SomeClass {
  private static var identifier = value;
}
```

Figure A.4 – A sample knowledge artefact created from a Personal Development Plan

Appendix B
User Evaluation Tasks

COOPERATIVE EVALUATION TASK

TASK INSTRUCTIONS

NOTE:

Below are lists of tasks for you to perform. While performing these tasks, please say aloud all the actions or tasks you are doing. You can say things like:

- "I am looking for/trying to do/...."
- "I want to/will like to/....."

Feel free to ask any question you may have.

TASK 1 ADDING AND REMOVING MEDIA COMPONENTS

1. Click the 'search twitter' icon in the Components toolbox
2. Type 'Manchester' in the search box, click 'search' or press enter to search
3. Resize the component to view the tweets comfortably
4. Move the component to a new position on the whiteboard
5. Double Click the word 'Enter Title' on the component title bar
6. Delete the word 'Enter Title'; type 'Tweet Box' and press enter
7. Repeat steps 5 and 6 if the word 'Enter Title' is still visible on the title bar
8. Minimise the component
9. Restore the component to its previous size
10. Minimise the component again
11. Click the 'Add Note' icon in the component toolbox to add a note component
12. Select the note component by clicking once on the title bar
13. Click on the edge icon that appears on top of the component and drag to touch the twitter component. (A connecting arrow should appear joining both components)
14. Repeat step 12 if the connecting arrow did not appear
15. Click on the arrow, type 'Tweet Box' in the textbox that appears
16. Click on a blank space to update the changes
17. Move the components around the whiteboard
18. Close the components to remove them from the whiteboard
19. Feedback session

TASK 2 CREATING A PERSONAL DEVELOPMENT PLAN

1. Click the PDP icon to add a PDP Component to the whiteboard
2. Populate the component Goal tab with the data provided in the task details sheet
3. Save the data
4. Populate the component Steps tab with the data provided in the task details sheet
5. Save the data
6. Repeat steps 1 – 6 using the alternative component (PDPlanner Type 2)
7. Feedback session

TASK 3 CONVERTING A PDP COMPONENT TO A KNOWLEDGE ARTEFACT

1. Using any one of the PDP components from the previous task
2. Add the specified components and link them to the PDP component (See the task details sheet)
3. Mark the first 3 steps on the PDP Steps tab as completed
4. Delete the last step on the PDP Steps tab
5. Minimise all the components including the PDP component
6. Arrange all the components neatly on the whiteboard
7. Feedback session

TASK 4 (RE) USING A PDP COMPONENT THAT HAS BEEN CONVERTED TO A KNOWLEDGE ARTEFACT

1. Inspect and review the PDP component provided for this session
2. Interact with the PDP component as specified in the steps on the PDP Steps tab
3. Feedback session

SIMULTANEOUS EVALUATION OF THE PERSONAL DEVELOPMENT PLANNING COMPONENT

NOTE:

This section of the evaluation is a simulation of a simultaneous session over the internet. Please do not forget to speak out as you perform the listed tasks. You will be working with another user in this session however; the other user will be working on another computer.

TASKS SUMMARY

1. Create a PDP Component
2. Collaboratively create a personal development plan
3. Convert the PDP Component to a learning artefact
4. Edit the Plan by removing some steps and components
5. Session Feedback

TASK DETAILS

In this session, you will be working with another user. One user will perform the role of ‘User A’, while the other will perform that of ‘User B’. Both users should perform their tasks independently except for task 1.

1. User A/B: Add a PDP component to the space.
2. User A: Fill in the PDP Goal
3. User B: Fill in the Success Criteria
4. User A: Fill in the target dates
5. User A: Save your data
6. User B: Save you data
7. User A and B: Select the PDP Steps tab on the component
8. User A: Add the provided steps
9. User B: Add the provided steps
10. User A: Save your data
11. User B: Save your data
12. User A: Add the provided components and link them to the PDP Component
13. User B: Add the provided components and link them to the PDP Component
14. User A: Interact with the components added by user B as specified
15. User B: Interact with the components added by user A as specified

The tasks are now complete for part one.

Part Two: Repeat the tasks described above using the alternative version of the PDP Component then proceed to the feedback session.

Appendix C
User Evaluation Consent Form

USABILITY IN USER GENERATED LEARNING SPACES

USER EVALUATION – SESSION CONSENT FORM

I agree to participate in the evaluation session detailed as follows:

- Conducted By: UDOISANG Blessing Sunday
- Designation: MSc Student, Advanced Computer Science & I.T. Management, School of Computer Science, University of Manchester
- Purpose: User evaluation for MSc Project Software
- Evaluation Type: Think Aloud; Cooperative Evaluation
- Maximum Duration: 20 Minutes

I understand and consent to the use of the following methods for recording the evaluation session. Please tick as applicable:

☐ Pen and Paper

☐ Computer Screen Capturing

☐ Audio Only Recording

☐ Video Only Recording

☐ Audio and Video Recording

I understand that the information and recordings is for research purposes only and that my name and image will not be used for any other purpose.

I relinquish any rights to the audio and video files (where applicable) and understand they may be copied and used for this research without further permission.

I understand that I can leave at any time.

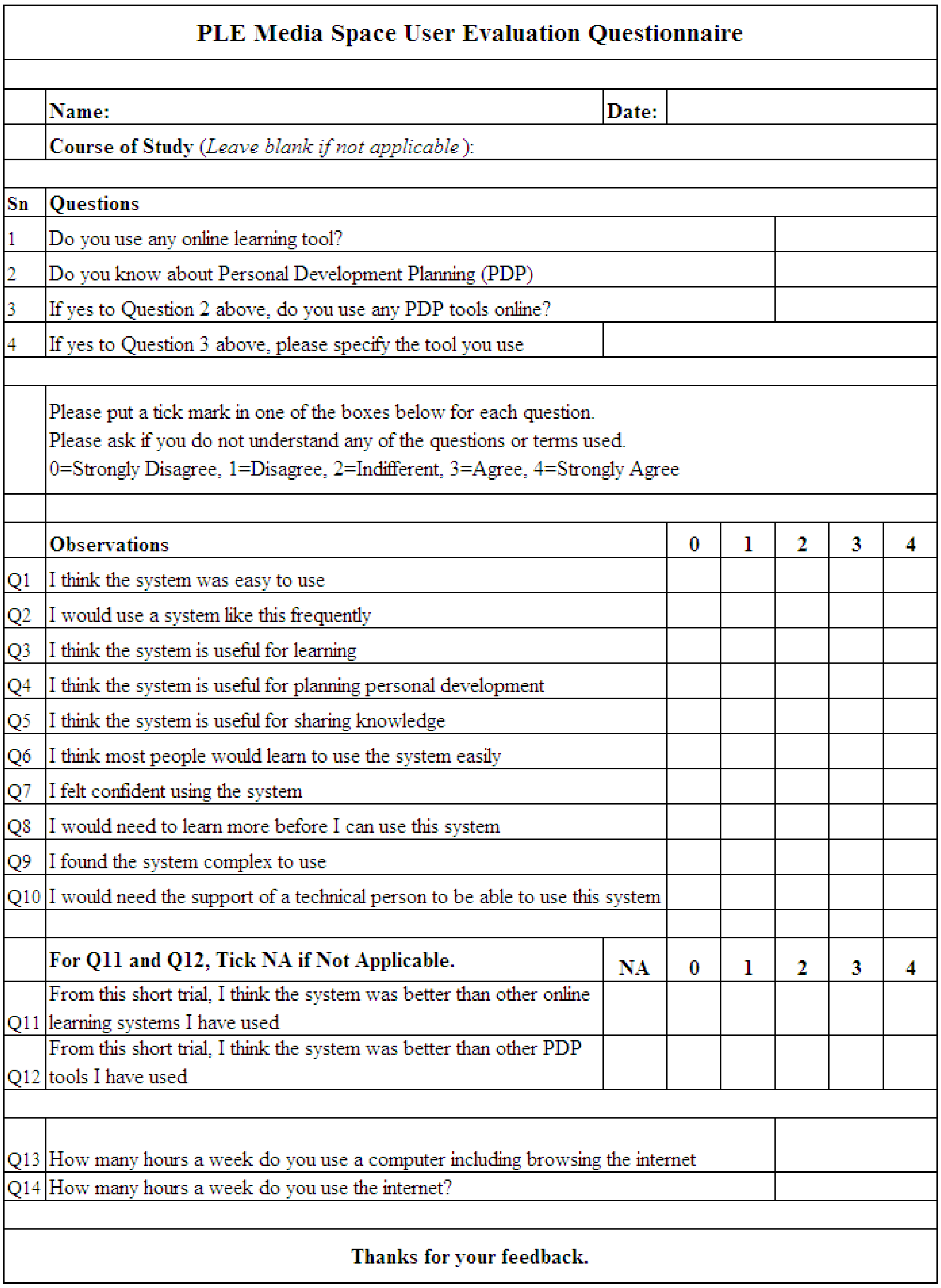
I agree to immediately raise any concerns or areas of discomfort with the session.

Please print your name: ___________________________________

Your signature and Date: ___________________________________

Thank you very much for participating, we appreciate it!

Appendix D
User Evaluation Questionnaire and Results



PLE Media Space User Evaluation Questionnaire

	Name:	**Date:**	
	Course of Study (*Leave blank if not applicable*):		

Sn	**Questions**	
1	Do you use any online learning tool?	
2	Do you know about Personal Development Planning (PDP)	
3	If yes to Question 2 above, do you use any PDP tools online?	
4	If yes to Question 3 above, please specify the tool you use	

Please put a tick mark in one of the boxes below for each question.
Please ask if you do not understand any of the questions or terms used.
0=Strongly Disagree, 1=Disagree, 2=Indifferent, 3=Agree, 4=Strongly Agree

	Observations	**0**	**1**	**2**	**3**	**4**
Q1	I think the system was easy to use					
Q2	I would use a system like this frequently					
Q3	I think the system is useful for learning					
Q4	I think the system is useful for planning personal development					
Q5	I think the system is useful for sharing knowledge					
Q6	I think most people would learn to use the system easily					
Q7	I felt confident using the system					
Q8	I would need to learn more before I can use this system					
Q9	I found the system complex to use					
Q10	I would need the support of a technical person to be able to use this system					

	For Q11 and Q12, Tick NA if Not Applicable.	**NA**	**0**	**1**	**2**	**3**	**4**
Q11	From this short trial, I think the system was better than other online learning systems I have used						
Q12	From this short trial, I think the system was better than other PDP tools I have used						

Q13	How many hours a week do you use a computer including browsing the internet	
Q14	How many hours a week do you use the internet?	

Thanks for your feedback.

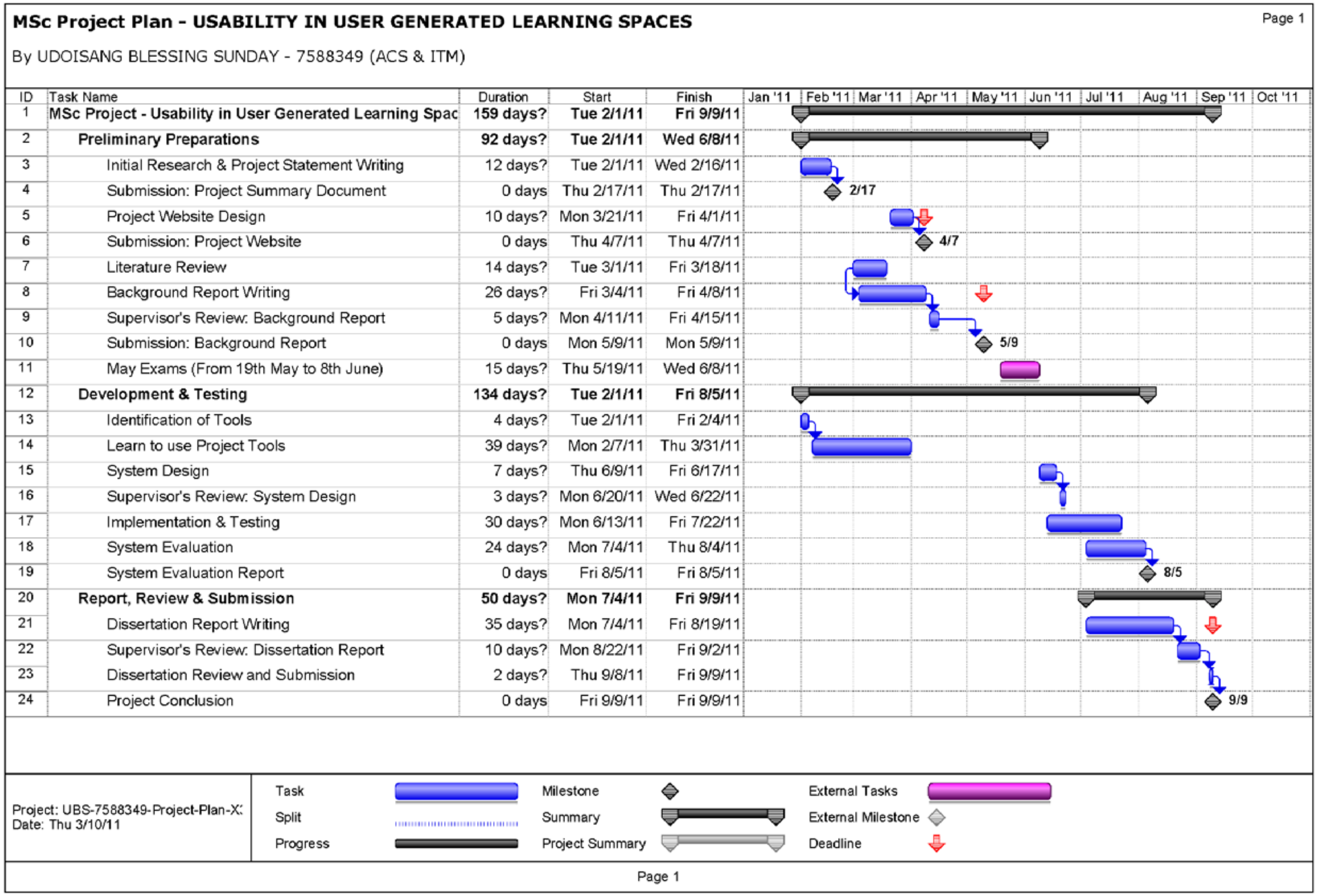
Questionnaire Results and the Calculated System Usability Score

Questions	P1		P2		P3		P4		P5		P6		P7		P8		P9		P10	
	Rw	Sc	Rw	Sc	Rw	Sc	Rw	Sc	Rw	Sc	Rw	Sc	Rw	Sc	Rw	Sc	Rw	Sc	Rw	Sc
Q1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	4	4	4	4	4	4
Q2	3	3	4	4	2	2	3	3	2	2	1	1	4	4	4	4	4	4	4	4
Q3	3	3	4	4	4	4	2	2	3	3	3	3	4	4	4	4	4	4	4	4
Q4	3	3	3	3	4	4	3	3	3	3	4	4	4	4	4	4	4	4	4	4
Q5	3	3	3	3	3	3	4	4	3	3	2	2	3	3	4	4	4	4	4	4
Q6	3	3	4	4	4	4	2	2	2	2	1	1	4	4	4	4	4	4	4	4
Q7	4	4	4	4	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4
Q8	2	2	0	4	1	3	1	3	0	4	3	1	1	3	0	4	0	4	0	4
Q9	0	4	0	4	0	4	3	1	0	4	1	3	0	4	0	4	0	4	0	4
Q10	0	4	1	3	0	4	1	3	0	4	3	1	0	4	0	4	0	4	0	4
CONTR		32.38		36.19		34.76		25.95		33.57		20.48		36.91		40		40		40
OVERALL		80.95		90.48		86.91		64.88		83.93		51.19		92.26		100		100		100
Average System Usability Score																				**85.061225**

Pn: Participant n **Rw**: Raw score obtained from participant **Sc**: Contributed score based on the scoring method (see Table 5 above)
CONTR: Sum of Contributions **OVERALL**: Overall usability score

Appendix E

Project Plan Gantt Chart



MSc Project Plan - USABILITY IN USER GENERATED LEARNING SPACES

Page 1

By UDOISANG BLESSING SUNDAY - 7588349 (ACS & ITM)

ID	Task Name	Duration	Start	Finish
1	**MSc Project - Usability in User Generated Learning Spac**	**159 days?**	**Tue 2/1/11**	**Fri 9/9/11**
2	**Preliminary Preparations**	**92 days?**	**Tue 2/1/11**	**Wed 6/8/11**
3	Initial Research & Project Statement Writing	12 days?	Tue 2/1/11	Wed 2/16/11
4	Submission: Project Summary Document	0 days	Thu 2/17/11	Thu 2/17/11
5	Project Website Design	10 days?	Mon 3/21/11	Fri 4/1/11
6	Submission: Project Website	0 days	Thu 4/7/11	Thu 4/7/11
7	Literature Review	14 days?	Tue 3/1/11	Fri 3/18/11
8	Background Report Writing	26 days?	Fri 3/4/11	Fri 4/8/11
9	Supervisor's Review: Background Report	5 days?	Mon 4/11/11	Fri 4/15/11
10	Submission: Background Report	0 days	Mon 5/9/11	Mon 5/9/11
11	May Exams (From 19th May to 8th June)	15 days?	Thu 5/19/11	Wed 6/8/11
12	**Development & Testing**	**134 days?**	**Tue 2/1/11**	**Fri 8/5/11**
13	Identification of Tools	4 days?	Tue 2/1/11	Fri 2/4/11
14	Learn to use Project Tools	39 days?	Mon 2/7/11	Thu 3/31/11
15	System Design	7 days?	Thu 6/9/11	Fri 6/17/11
16	Supervisor's Review: System Design	3 days?	Mon 6/20/11	Wed 6/22/11
17	Implementation & Testing	30 days?	Mon 6/13/11	Fri 7/22/11
18	System Evaluation	24 days?	Mon 7/4/11	Thu 8/4/11
19	System Evaluation Report	0 days	Fri 8/5/11	Fri 8/5/11
20	**Report, Review & Submission**	**50 days?**	**Mon 7/4/11**	**Fri 9/9/11**
21	Dissertation Report Writing	35 days?	Mon 7/4/11	Fri 8/19/11
22	Supervisor's Review: Dissertation Report	10 days?	Mon 8/22/11	Fri 9/2/11
23	Dissertation Review and Submission	2 days?	Thu 9/8/11	Fri 9/9/11
24	Project Conclusion	0 days	Fri 9/9/11	Fri 9/9/11

Project: UBS-7588349-Project-Plan-X:
Date: Thu 3/10/11

Task	▬	Milestone	◆	External Tasks	▬
Split	┄	Summary	▬	External Milestone	◇
Progress	▬	Project Summary	▬	Deadline	⇩

Page 1