A Framework for the Assembly and Delivery of Multimodal Graphics in E-Learning Environments

Declan McMullen

Bachelor of Science in Computer Science

Master of Science in Computer Science

A Dissertation submitted in fulfilment of the

requirements for the award of

Doctor of Philosophy (Ph.D.)

to the

Dublin City University

Faculty of Engineering and Computing, School of Computing

Supervisor: Dr Donal Fitzpatrick

April 2011

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed:

ID No.: 50721204

Date: April 18, 2011

Contents

Abstract

In recent years educators and education institutions have embraced E-Learning environments as a method of delivering content to and communicating with their learners. Particular attention needs to be paid to the accessibility of the content that each educator provides. In relation to graphics, content providers are instructed to provide textual alternatives for each graphic using either the "alt" attribute or the "longdesc" attribute of the HTML IMG tag. This is not always suitable for graphical concepts inherent in technical topics due to the spatial nature of the information. As there is currently no suggested alternative to the use of textual descriptions in E-Learning environments, blind learners are at a significant disadvantage when attempting to learn Science, Technology, Engineering or Mathematical (STEM) subjects online. A new approach is required that will provide blind learners with the same learning capabilities enjoyed by their sighted peers in relation to graphics.

Multimodal graphics combine the modalities of sound and touch in order to deliver graphical concepts to blind learners. Although they have proven successful, they can be time consuming to create and often require expertise in accessible graphic design. This thesis proposes an approach based on mainstream E-Learning techniques that can support non-experts in the assembly of multimodal graphics. The approach is known as the Multimodal Graphic Assembly and Delivery Framework (MGADF). It exploits a component based Service Oriented Architecture (SOA) to provide non-experts with the ability to assemble multimodal graphics and integrate them into mainstream E-Learning environments.

This thesis details the design of the system architecture, information architecture and methodologies of the MGADF. Proof of concept interfaces were implemented, based on the design, that clearly demonstrate the feasibility of the approach. The interfaces were used in an end-user evaluation that assessed the benefits of a component based approach for non-expert multimodal graphic producers.

Acknowledgements

I wish to thank my supervisor Dr. Donal Fitzpatrick for his support and encouragement during this research. He has always made himself available for advice and remained confident that I could complete the work in spite of numerous setbacks.

I am grateful to the European Union Socrates Minerva Fund and Dublin City University's Learning Innovation Unit for funding elements of this research. The work could not have been completed without their financial support.

To all of the members of the School of Computing postgraduate community with whom I have had the pleasure of spending the last number of years, I express my sincere thanks. Particularly those who have resided at desks in bays A and B. You have always been available to provide knowledge, humour and a friendly ear. I also wish to extend my gratitude to Dr. Mark Melia and Dr. Michelle Graham who provided invaluable feedback on the content of this thesis.

Particular thanks must go to those who gave their time in order to take part in the evaluation phase of this research. Also, I express my gratitude to Stuart Lawler and Joshue O Connor from the NCBI who have always been extremely gracious with their time, knowledge and guidance.

My sincere gratitude goes to Dublin City University's student support services. In particular I wish to thank Hadas Levy for supporting me through some difficult times and helping me to become a stronger person.

To Una, I cannot thank you enough for believing in me even at times when I did not. Your support and understanding has driven me to complete this work no matter what obstacles needed to be overcome. I love you and I hope to reach the potential you believe I have.

To my parents, who have supported me emotionally and financially throughout my life, I am eternally grateful. You have never allowed me to give up and always believed that I was capable of achieving anything I desired. You have given me everything I need in life to succeed and I hope to make you proud.

This thesis is dedicated to the memory of my grandfather Cyril McMullen

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

In recent years educators and education institutions have embraced E-Learning environments as a method of delivering content to and communicating with their learners. Numerous environments now exist including open source implementations, such as Moodle [Moodle, 2010] and commercial applications, for example Blackboard [Blackboard, 2010]. E-Learning can be used as the primary method of providing content to a learner, such as those enrolled in distance education programmes or it can be part of an overall blended learning scenario where physical lectures are supplemented with online content. Regardless of how an E-learning environment is used, particular attention needs to be paid to the accessibility of the content it provides. In April 2008, a report commissioned by the Association on Higher Education and Disability (AHEAD) found that blind/vision impaired learners are 50% less likely to progress from second level to third level education and are significantly disadvantaged in comparison to their non-disabled classmates [AHEAD, 2008]. A large factor in this disparity is a lack of equality in terms of access to learning content.

The problem has been solved to an extent where textual information is concerned. Content can be provided using interactive audiobooks [DAISY, 2010], screenreaders [JAWS, 2010] [GWMicro, 2010] and refreshable Braille displays [HumanWare, 2010]. Blind learners therefore have relatively equal access to textual information and can benefit from the independent learning opportunities an E-Learning environment can provide.

This is not the case in relation to graphical content [Gardner and Bulatov, 1997]. When placing content on the Internet, content providers are advised to conform to a set of instructions that aim to ensure all online content is accessible to those with a disability. These instructions are called the Web Content Accessibility Guidelines (WCAG) [WAI, 2008] published by the World Wide Web Consortium's Web Accessibility Initiative (W3C-WAI) [WAI, 2010]. Guideline 1.1 of WCAG 2.0 relates to the use of graphics on the web and states that those placing graphics on the web should, "provide text alternatives for any non-text content so that it can be changed into other forms people need ... ". As E-Learning environments are built on top of web technologies, the instructions given to those creating lesson content has been to follow guideline 1.1 of WCAG and provide text alternatives for each graphic using either the "alt" attribute or the "longdesc" attribute of the HTML IMG tag [E-Learn-VIP, 2010] [Fisseler and Bühler, 2007]. The "longdesc" attribute has been most appropriate as it can provide more detailed descriptions of complex images.

Therefore, an inequality arises between the learning experience of sighted learners and those with a visual disability. A sighted learner will have access to graphics in order to supplement their learning and aid in the explanation of concepts that require an understanding of issues such as spatial relationships, connections between elements, scale and so on. This type of graphically reinforced learning is particularly important when technical topics are being discussed. It is difficult to master technical topics, that are by their nature graphically intensive, using only textual descriptions [Gardner and Bulatov, 2006]. When textual descriptions are used, blind learners are not able to gather any impression about the shape and layout of graphics [Kraus et al., 2008].

As there is currently no suggested alternative to the use of textual descriptions in E-Learning environments, blind learners are at a significant disadvantage when attempting to learn Science, Technology, Engineering or Mathematical (STEM) subjects online. Additionally, at time of writing, the "longdesc" attribute has been removed from the emerging HTML 5 standard. Therefore, next generation E-Learning content will be unable to use the "longdesc" attribute and remain compliant, thus removing the recommended method of graphic accessibility for blind learners. It is therefore more important than ever that a new approach be designed that will provide blind learners with the same learning capabilities enjoyed by their sighted peers in relation to graphics.

1.2 Research Context

Solutions exist that can provide blind learners with access to graphical material but they operate externally and independently of E-Learning environments. The most common solution is that of “tactile graphics”. A tactile graphic is a two dimensional image that is provided on a special sheet of paper or plastic. The image has a depth of a few millimetres that rises upwards from the paper and allows the learner to explore the graphic using only their fingers. This form of tactile exploration can provide a learner with knowledge of shapes and spatial relationships that relate to the topic depicted on the graphic [Edman, 1992].

As the human eye has a much higher resolution than the human finger, the same images that are provided to sighted learners cannot be directly provided to blind learners for tactile exploration [Gardner, 1996]. Tactile graphics are simplified versions of graphical concepts and must contain less detail and more space in order for a learner to understand them. A comparison between a human eye mainstream graphic and a human eye tactile graphic can be seen in figures 1.1 and 1.2.

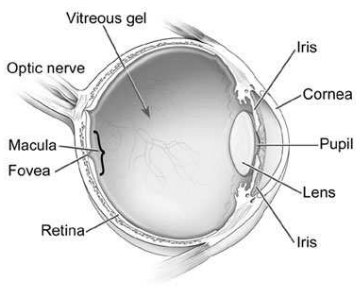


Figure 1.1: Graphic of the human eye

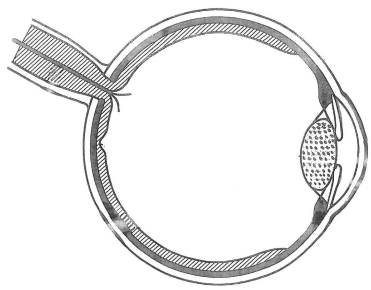


Figure 1.2: Tactile graphic of the human eye

A level of expertise in tactile design is generally required if a teacher plans to use tactile graphics in their course material. Additionally, supplementary information is always required alongside a tactile graphic. A learner will not be able to understand what is depicted on a tactile graphic without some background information and data relating to each individual element of the graphic that a learner can explore [Ungar et al., 1997]. Supplementary information is generally provided by a sighted guide or via Braille labelling. This form of tactile graphic delivery does not have a digital form and therefore is not suitable for use in E-Learning environments. A digital approach exists that may provide a viable solution for graphics in E-Learning.

1.2.1 Audio Tactile Graphics

A reliance on Braille labelling in tactile graphics contains numerous drawbacks. As there is limited space on a graphic, only a small number of labels can be placed on it [Miele and Schaack, 2008]. This restricts the amount of supplementary information that can be provided to the learner [McMullen and Fitzpatrick, 2009]. The location of the labels can cause confusion for a learner as they may struggle to identify which element of the graphic a specific label refers to [Aldrich and Sheppard, 2001]. If you are not a Braille reader, the labels are of no use to you [Miele and Schaack, 2008] [Landau and Gourgey, 2001]. A technology, known as audio tactile graphics, aims to provide solutions to the problems outlined above.

Audio tactile graphics involve placing a tactile graphic on top of a touch sensitive screen that is connected to a computer. When the learner presses on an element of the tactile graphic, supplementary information, relating to the element that was pressed, is played back through the computers speakers in either synthetic speech or pre-recorded audio.

The primary benefit of an audio tactile is the ability to have multiple layers of auditory information for each element of the graphic. This is in contrast to Braille labelling, where a shortage of space caused the provision of limited supplementary information. In addition, audio is accessible by all who wish to avail of audio tactile technology, thus removing the access restrictions previously in place requiring learners to possess Braille literacy skills. As the delivery environments that provide access to audio tactiles are software based and all of the content is stored digitally, audio tactiles are a viable technology for providing graphics to blind learners in E-Learning environments.

Although audio tactiles are a viable technology, they currently operate entirely independently of E-Learning platforms. Audio tactiles are not integrated into any form of lesson structure and therefore their graphical content is used individually and in isolation. In order for an instructor to provide audio tactile graphics to their blind learners, they must make them available for delivery in external environments. This involves maintaining separate systems for blind and sighted learners which increases the workload on the instructor. Therefore, an approach is required where audio tactiles can be delivered in mainstream E-Learning environments alongside content for sighted learners.

In order to accomplish this, the approach must provide non-expert audio tactile producers with an intuitive method of audio tactile construction so that they can provide graphics as easily to their blind learners as they do to their sighted learners. Historically, instructors tended to avoid using tactiles or audio tactiles in their lessons as their creation was seen as extremely time consuming [Sheppard and Aldrich, 2001]. This is an issue that must be avoided if audio tactiles are to find their way into mainstream E-Learning environments.

Technology Enhanced Learning (TEL) is the term used to describe the support of a learning activity through the use of technology. Research in the area deals primarily with content management, courseware construction and courseware delivery. Content reusability frameworks provide course creators with access to reusable learning objects. Learning objects are units of content that can be shared amongst courseware creators to reduce the construction time of new courseware. This approach allows a courseware creator to operate at a high level of abstraction thereby facilitating simple course construction. Research into standardisation provides for a course, once constructed, to be delivered in numerous different learning environments. This allows a course creator to provide learning content to numerous learners using multiple different environments with minimal effort. The techniques described above coupled with approaches to information and system architectures may provide the key to non-expert audio tactile creation.

The approach in this thesis focusses on graphics that are delivered using a range of modalities consisting of on screen visuals, touch interaction and audio feedback. As multiple forms of each modality can be supported, for example tactile and haptic touch interaction, the term multimodal graphic is used from this point on to represent graphics that conform to the description above. Additional modalities such as those defined in the W3C Extensible MultiModal Annotation markup language (EMMA)[W3C, 2009] are not considered at this time. It should be noted that although multimodal graphics can be used by sighted learners, this thesis focuses on their use by blind learners.

1.3 Research Questions

Siqueira et al stated that "e-learning initiatives should consider an e-learning infrastructure, which should be based on an e-learning architecture, which, in turn, should be based on a generic approach" [Siqueira et al., 2008]. Any approach that aims to allow for the integration of multimodal graphics in E-Learning environments should therefore be based on mainstream E-Learning techniques. As such the research question defined for this multi-disciplinary applied research is, "Can mainstream E-Learning concepts relating to componentisation, information architecture and system architecture, be applied in an approach to multimodal graphic assembly"? The primary research question raises two sub questions that must be investigated, is the approach, once designed, feasible and does it provide benefits to the assembler of multimodal graphics?

1.4 Goals and Objectives

There are numerous goals that must be satisfied in order to complete the research. The approach suggested in this work should:

1. Facilitate simple creation of multimodal graphics by non-experts in multimodal graphic assembly.
2. Facilitate faster creation of multimodal graphics when exploiting reusability.
3. Be capable of being integrated into mainstream E-Learning platforms.
4. Maintain a separation between graphical components such that they are independently reusable.
5. Facilitate the assembly and delivery of multimodal graphics and their components in multiple environments.
6. Be capable of supporting various types of visual and non visual content combinations.

7. Be extensible in order to support additional functionality.

In order to fulfil the goals set out above, numerous objectives must be reached. These objectives are to:

1. Research and identify specific techniques in the areas of componentisation, information architecture and system architecture, that can be applied in the area of multimodal graphic assembly.
2. Identify the independent components that form a multimodal graphic.
3. Design an extensible architecture capable of supporting the independent components of a multimodal graphic and allowing for their interaction.
4. Design data models that can be used to represent the independent components of a multimodal graphic and provide interoperability.
5. Design a methodology for combining independent components into a coherent[1] multimodal graphic.
6. Implement a proof of concept system that clearly demonstrates the feasibility of the approach.
7. Evaluate the system with non-expert multimodal graphic assemblers to assess the benefits of the approach.

1.5 Thesis Overview

The rest of the thesis is arranged as follows. Chapter 2 provides a background to the area of Technology Enhanced Learning (TEL). Specific attention is paid to the areas of componentisation, system architecture, information architecture and interoperability. Research in the area of learning objects, learning object repositories, Adaptive Educational Hypermedia (AEH) and standardisation are discussed. Chapter 3 provides an overview of the state of the art for providing blind learners with access to graphical data. The majority of the chapter

[1]Further explanation of the term coherent, as it relates to this research, can be found on page 55

focuses on the production and usage of tactile graphics. Audio tactile interfaces are discussed and two commercial approaches contrasted. Emerging interfaces are discussed and the chapter ends with a review of issues that remain to be solved. Chapter 4 discusses the Multimodal Graphic Assembly and Delivery Framework (MGADF), an approach designed to harness mainstream techniques to provide non-experts with simple multimodal graphic assembly capabilities. The system architecture, information architecture and methodologies of the approach are described. Chapter 5 describes the implementation of proof of concept systems that assess the feasibility of the approach. The technologies used in the implementations are discussed along with the integration of the methodologies in a number of user interfaces. Chapter 6 outlines the design and implementation of a summative evaluation using the proof of concept tools discussed in chapter 5. The information architecture is evaluated using a use case and a scenario based evaluation method is used to assess the system architecture. The results of the evaluation are compared to the goals, objectives and research questions defined in chapter 1. The thesis ends with chapter 7 where the work is summarised, remaining problems discussed and future work outlined.

Chapter 2

Background, Technology Enhanced Learning

2.1 Introduction

Technology Enhanced Learning (TEL) is a vast and constantly evolving research area. Architecture design, content management, standardisation, personalisation, courseware construction and instructional design are but a few of the topics that exist within the area of TEL. As not all areas of TEL are relevant to our research, this chapter will focus on the areas that influenced the direction of the work. Elements of this chapter will reappear in chapter 4 in relation to how they influenced the design of our suggested approach to multimodal graphic assembly. Section 2.2 discusses the area of component based courseware construction. The section focusses on the use of reusable learning objects and outlines common techniques and methodologies. Various approaches to Information Architecture design are discussed in section 2.3. Adaptive systems and their use of multiple independent models are described. Section 2.4 relates to System Architecture. Standardisation efforts and the use of Service Oriented Architectures (SOA) are discussed. The chapter ends in section 2.5 with an overview of interoperability. Various approaches to data standardisation and content packaging are outlined.

2.2 Componentisation

2.2.1 Learning Objects

Educators in numerous international institutions were teaching the same topics on a regular basis. Although the content for those topics was predominantly the same, each educator was developing new material to meet the needs of their course. Savings could be made in terms of time and cost if courseware could be reused and shared amongst the education community [Wiley, 2000]. A solution was designed based on the object-oriented paradigm of computer science. In object-oriented programming, primary components of a systems functionality can be defined as "objects". These objects can then be independently reused and composed in order to construct a complete system [Wiley, 2000]. The educational equivalent of these components are called "learning objects". Learning objects are small reusable chunks of learning content that can be aggregated and sequenced into a coherent lesson. A lesson is a combination of learning material that is designed to facilitate the completion of specific learning objectives. Learning objects have been defined as, "any entity, digital or non-digital, which can be used, re-used or referenced during technology supported learning" [IEEE, 2002]. It is possible for an educator to create and share learning objects in a simple manner by making content available in standard formats. A Powerpoint presentation or a Word document containing instructional material can be considered a learning object, which can be reused by various educators and provided to their learners.

In order for an approach based on reusable content to be beneficial, a mechanism was required that would allow educators to search for relevant content. A large repository of content was of limited use without the ability to classify and index the information to support it's discovery and reuse. In order to achieve that goal, a mechanism for associating descriptive data with learning resources was designed. Three standards emerged for adding descriptive information to content, the IMS Learning Resource Metadata (LRM) Information Model [IMS, 2001], IEEE Learning Object Metadata (LOM) [IEEE, 2002] and the Dublin Core Metadata Initiative (DCMI) [DCMI, 2004]. As LRM has been superseded by LOM, it will not be discussed further. A brief overview of LOM and Dublin Core is now provided.

LOM can be used to "facilitate search, evaluation, acquisition and use of learning objects" [IEEE, 2002]. It consists of 9 categories of information; General, Life Cycle, Meta-Metadata, Technical, Educational, Rights, Relation, Annotation and Classification. These categories allow a learning object creator to provide details such as keywords, language, format, difficulty level, copyright information and so on. Metadata can be used to facilitate searching, allowing educators to locate suitable learning objects that they can reuse in their own courses.

Dublin Core is a non E-Learning specific metadata standard. It provides a common core of semantics for resource description that can also aid in content discovery and facilitate interoperability. There are 15 Dublin Core elements [DCMI, 2010b];

Title A name given to the resource.

Creator An entity primarily responsible for making the resource.

Subject The topic of the resource.

Description An account of the resource.

Publisher An entity responsible for making the resource available.

Date A point or period of time associated with an event in the lifecycle of the resource.

Type The nature or genre of the resource.

Identifier An unambiguous reference to the resource within a given context.

Language A language of the resource.

Contributor An entity responsible for making contributions to the resource.

Format The file format, physical medium or dimensions of the resource.

Source A related resource from which the described resource is derived.

Relation A related resource.

Coverage The spatial or temporal topic of the resource.

Rights Information about rights held in and over the resource.

2.2.2 Learning Object Repositories

Creating a learning object and associating descriptive metadata does not in itself ensure that a wide range of educators can make use of it. Learning objects must be stored in a specific location and functionality provided that allows educators to search for and retrieve learning material that meets their needs. Learning object repositories emerged to fill this role by providing a facility where learning objects and their metadata could be stored. Numerous learning object repositories exist internationally including, MERLOT (Multimedia Educational Resource for Learning and Online Teaching) [MERLOT, 2010], ARIADNE [ARIADNE, 2010], NDLR (National Digital Learning Resources) [NDLR, 2010], GEM (Global Entrepreneurship Monitor) [GEM, 2010], LOLA (Learning Object: Lifelong Application) [LOLA, 2010], and GESTALT (Getting Educational Systems Talking Across Leading-Edge Technologies) [Wade and Doherty, 2000]. Repositories often allow educators to store learning content aswell as the metadata that relates to it, for example the NDLR. Other repositories act as metadata only repositories that point to the location where the content can be found, such as MERLOT. In most cases users can search for and retrieve learning objects without requiring special privileges. In order to store content, an account with additional access rights is generally required. This maintains the integrity of the repository and aids in ensuring that learning objects are of high quality.

2.2.3 Courseware Construction

Courseware construction environments, which allow course creators to benefit from reusable learning objects, are referred to as content reusability frameworks [Brusilovsky et al., 2008]. These frameworks allow the course creator to operate at a high level of abstraction. Rather than creating all of their learning content, they can search for, retrieve and sequence existing content into coherent courseware. This turns an educator into a content assembler rather than a content creator. The existence of these frameworks avoids the need for educators to redevelop the same material.

The traditional courseware reuse model begins with the educator specifying the content they require in terms of its attributes, such as pedagogical type, topic or duration. A search is performed in a repository for content that conforms to the specification. Once a suitable learning object has been identified, the educator can retrieve it and insert it into their course. The reusability approach reduces course development time and improves course quality [Brusilovsky et al., 2008].

2.3 Information Architecture

Research has pointed to weaknesses in content reusability frameworks, primarily in the area of "one size fits all" instruction [Brusilovsky et al., 2008] [Conklin, 1987] [Dagger et al., 2005]. Learning objects may be used by learners with different backgrounds, abilities and learning styles. During the authoring phase, the learning object creator must design generic content that will be usable by all learners regardless of preference or ability. This can impact on a students ability to learn, as the material is not delivered to them in a suitable fashion [Dagger et al., 2005] [Meister, 2002] [Frankola, 2001].

2.3.1 Adaptive Educational Hypermedia

Adaptive Educational Hypermedia (AEH) systems were developed in order to solve the "one size fits all" problem. They are based on technologies in the area of Adaptive Hypermedia (AH) [Brusilovsky, 1996] and Intelligent Tutoring System (ITS) [Polson and Richardson, 1988]. An AEH system is capable of providing "just in time" delivery of personalised learning content [Brady et al., 2006]. This means that it can dynamically generate personalised learning content designed to meet specific user needs. The axes of personalisation can include prior knowledge, competencies, learning styles, language preferences and learning goals. In order for an AEH system to provide personalised functionality, data relating to the axes of personalisation must be modelled. The user model can be populated using a questionnaire [Felder and Silverman, 1988], by means of a pre-test, via constant student monitoring or via all of the above.

AEH systems consist primarily of adaptive navigation and adaptive presentation techniques [Froschl, 2005]. As there are a number of diverse elements involved in a successful AEH system, researchers turned to a model based approach to support the authoring and delivery of personalised courseware. Some examples are provided in the following sections.

2.3.2 Adaptive Hypermedia Authoring Model

The Adaptive Hypermedia Authoring Model (AHAM) is a generalised information model for applying adaptive hypermedia [DeBra et al., 1999] [DeBra et al., 2002]. It is based on the Dexter hypertext reference model [Halasz and Schwartz, 1994] and consists of storage, runtime and within component layers. AHAM extends the storage layer of Dexter by including a user model and a teaching model. The teaching model consists of pedagogical rules that can query the user model in order to produce a presentation specification. This specification is generated by adapting the domain model in accordance with the contents of the user model. The layers of AHAM can be seen in figure 2.1.

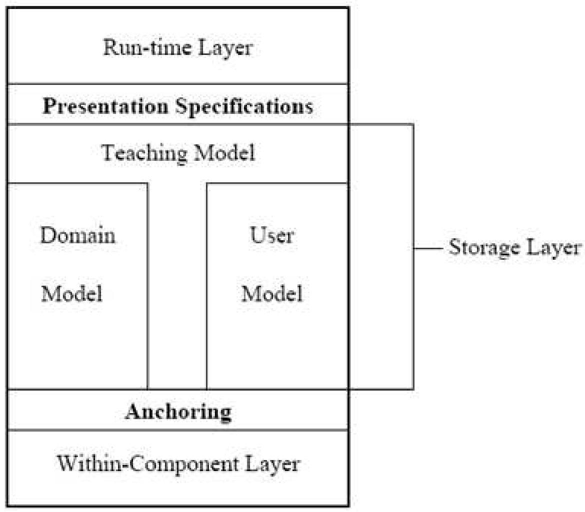


Figure 2.1: AHAM Layers

2.3.3 LAOS

LAOS (Layered Adaptive hypermedia authoring model and its algebraic OperatorS), is a system of layered models which extends AHAM [Cristea and de Mooij, 2003]. There are two types of models, static and dynamic, arranged in five layers built on top of one another. LAOS contains a Domain Model, Goals and Constraints Model, User Model, Presentation Model and Adaptation Model. The static models contain information relating to the learner, the domain and the pedagogy. The dynamic model describes how a system should adapt to variations in the static models. The domain knowledge for a course is defined in terms of concepts. Attributes of the concepts can point to specific learning content. These concepts are organised and structured in the Domain Model. The Goals and Constraints model represents educational goals by assigning weights to the domain concepts. It can also affect the constraints of the adaptation based on the User Model. The User Model defines a learner's knowledge, learning style and educational goals. It contains a concept map of user variables and their values. The Adaptation Model consists of a series of condition/action rules, which define how the approach should react to the data in the static models. The adaptation model influences the pedagogical strategy used in the final course delivery. The Presentation Model details the physical properties of a delivery environment. This model can be used to adapt the contents of a course to suit different delivery environments. The layers of the LAOS model can be seen in figure 2.2. LAOS has been successfully exploited as a basis for the "My Online Teacher (MOT)" AEH system [Cristea et al., 2005].

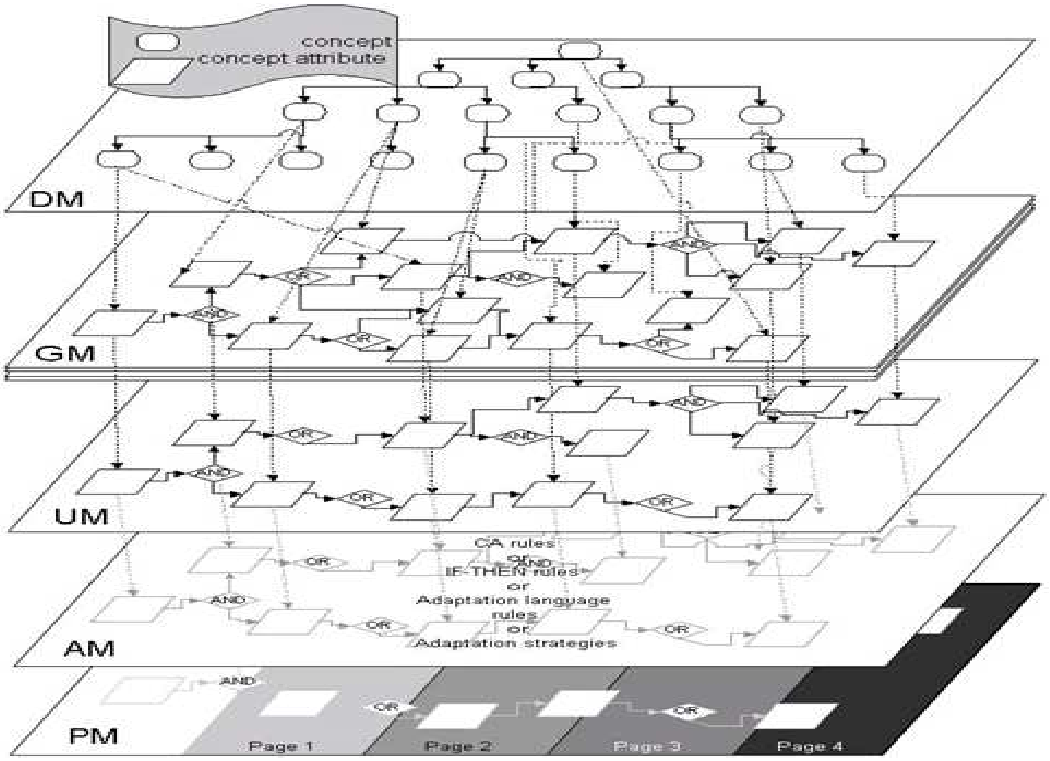


Figure 2.2: LAOS Layers

2.3.4 Multi-Model, Metadata Driven Approach

The Multi-Model, Metadata Driven approach to Adaptive Hypermedia is based on a core methodology of abstraction [Conlan et al., 2006] [Conlan, 2004] [Conlan et al., 2002]. The hypothesis is that separating the concerns of adaptivity and E-Learning makes it easier to reuse content outside of it's original context [Dagger et al., 2003]. The approach consists of three core models, the Content Model, the Learner Model and the Narrative Model. The Content Model contains metadata that describes a learning resource. The metadata can contain information such as the pedagogical and technical attributes of a given piece of content. The Learner Model is an overlay model containing sections of the domain that are currently being taught, as specified in the narrative. The Narrative Model contains sequencing rules and metadata that represent the various possibilities that an adaptive engine can use to personalise the learning experience. Narratives are generally developed by domain and pedagogical experts. This approach allows for the various sequences that can be used to deliver a course to be separated from the content itself. Figure 2.3 illustrates the Multi-Model Metadata Driven approach. The approach has been successfully used as a basis for the Adaptive Personalised E-Learning Service (APeLS)[Conlan and Wade, 2004].

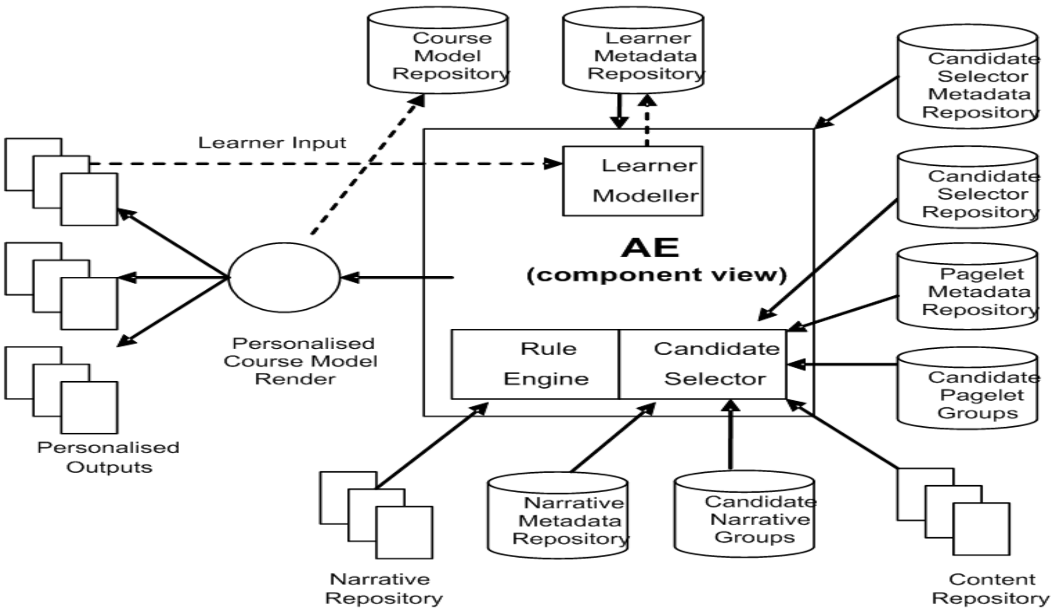


Figure 2.3: Multi-Model Component Architecture

2.4 System Architecture

Architectures define structures that connect an E-Learning system as a software and information system with its instructional and educational context [Bouras and Tsiatsos, 2006].

A lot of standardisation efforts have focussed on the description of learning resources leading to enhanced learning content interoperability. Research has been limited however in the area of E-Learning architecture standardisation.

2.4.1 Standardisation

The IEEE Learning Technology Systems Architecture (LTSA) [IEEE, 2001] specifies a high level reference architecture for E-Learning systems. The specification identified the objectives of human activities, computer processes and categories of knowledge that exist in an E-Learning scenario. No specific details relating to implementation technologies for creating system components or management systems for maintaining content resources are provided in the standard.

The Content Object Repository Discovery and Registration/Resolution Architecture (CORDRA) [Rehak et al., 2005] was developed by the Learning Systems Architecture Lab at Carnegie Mellon University [LSAL, 2010]. CORDRA is an "open, standards based model for how to design and implement software systems for the purpose of discovery, sharing and reuse of learning content through the establishment of interoperable federations of learning content repositories" [Rehak et al., 2005]. The main activity was the definition of a reference model designed to be an enabling bridge between learning content management/delivery and digital libraries. The model provides guidelines and standards for how to design and implement content repositories based on CORDRA in order to facilitate a single point of learning object discovery.

The E-Learning Framework (ELF) project aimed to provide a common vocabulary and roadmap for the development of component services in an E-Learning infrastructure [ELF, 2010]. In the framework, services are classified as E-Learning specific services or common services. Course management, for example, is an E-Learning specific service, whereas authentication is a common service. ELF promotes the use of standards and specifications in the area of E-Learning and promotes the implementation of open source toolkits.

2.4.2 Generic Architecture

As E-Learning systems can consist of a number of components performing different tasks and as a number of architectural approaches are available for supporting those tasks, a generic E-Learning architecture was designed [Siqueira et al., 2003]. The architecture is not oriented towards a specific learning approach but instead aims to provide all possible components of an E-Learning system. The architecture can be seen in figure 2.4. Core components of E-Learning functionality are specified and additional operations that relate to those components are identified. For example, in relation to Content Development, the architecture illustrates that this may involve; Sequencing, Media Selection, Authoring, Composition and Metadata Edition. The Content Selection service of the generic architecture was successfully integrated into the design of the LORIS architecture [de Moura et al., 2005]. LORIS (Learning Objects Repositories Integration System) has been further extended to cater for the selection of learning objects based on a user model, AccessForAll-LORIS [Ghelman et al., 2006], and for integrating learning objects with digital library repositories, LORDiLIS (Learning Objects Repositories and Digital Libraries Integration System) [Gomes et al., 2005].

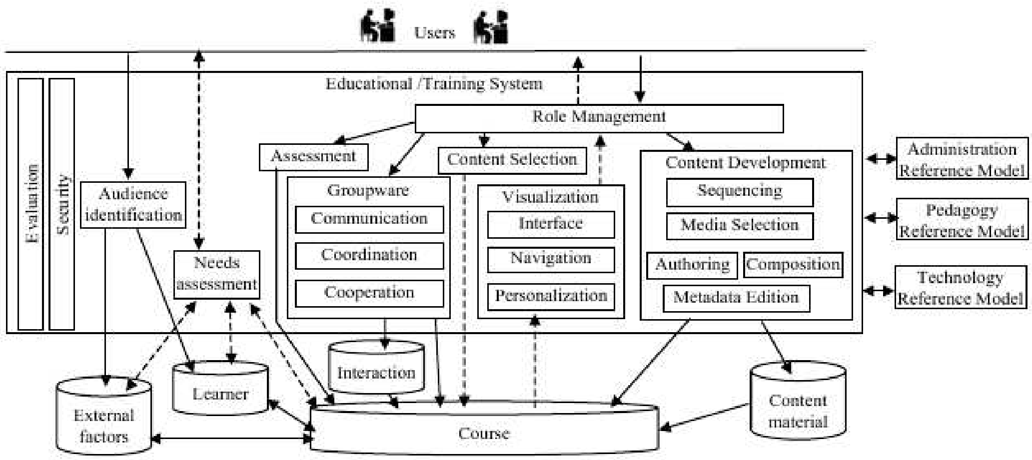


Figure 2.4: Generic E-Learning Architecture

2.4.3 Service Oriented Architectures

The recent generation of E-Learning systems, mostly in the AEH area, are based on model separation and Service Oriented Architectures (SOA) [Brusilovsky et al., 2008]

[Brusilovsky, 2004a]. Web services provide interoperability among components as they are based on open data standards and communication protocols such as the Simple Object Access Protocol (SOAP) [W3C, 2007a] and the Web Services Description Language (WSDL) [W3C, 2007b]. A web service is defined by the W3C as "a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL)." [W3C, 2004a]. E-Learning systems based on web services can contain content services, presentation services, collaboration services and monitoring services that support the reuse of system components [Türker et al., 2006] [Brady et al., 2005] [Brady et al., 2006]. Some specific service oriented E-Learning systems are discussed in the following sections.

The Adaptive Personalised E-Learning Service (APeLS) [Conlan and Wade, 2004] is based on the multi-model metadata driven approach discussed previously. Rather than retrieving content from APeLS and copying it into a delivery environment, content is provided as a service. Content can be delivered using a portal or E-Learning system capable of communicating with the service. Three separate models exist in the service, the Learning Content Model, the Narrative Model and the Learner Model. An adaptive engine generates personalised courseware dynamically at runtime, based on the contents of the three models. The Content Model contains metadata descriptions of the learning objects. The Narrative Model contains the concepts that may be selected for delivery in a course. No direct link exists between learning objects and Narrative concepts. The individual components are reconciled at runtime by the adaptive engine. The Learner Model contains the preferences and abilities of the learner that the adaptive engine can use to personalise course delivery. The APeLS architecture is extensible, meaning that additional models can be developed and integrated into the service.

As the metadata for a component is separated from the component itself, APeLS consists of a number of different repositories. The Learner Metadata Repository stores the learner models for each user of the system. The Content Metadata Repository stores Content Models that point to concrete learning objects. The Narrative Metadata Repository contains a description of the Narratives that can be used to deliver a course using a variety of pedagogical strategies. The Content Repository stores the concrete learning objects that can be delivered to the learner. The Narrative Repository stores all available Narratives that can be used to structure a course. The APeLS architecture is based on the Multi-Model approach as seen in figure 2.3.

Another service oriented AEH system is KnowledgeTree [Brusilovsky, 2004b] [Brusilovsky, 2004a]. In comparison to APeLS which consists of a single service, KnowledgeTree consists of a number of communicating servers. Three kinds of servers are supported, activity servers, student model servers and learning portals. The KnowledgeTree architecture can be seen in figure 2.5. Learning portals follow the same theory as in APeLS, where a lightweight portal provides access to content that resides on a server. Course creators can use the portal to design their courseware and deliver the content to their users. The content can reside in multiple distributed activity servers.

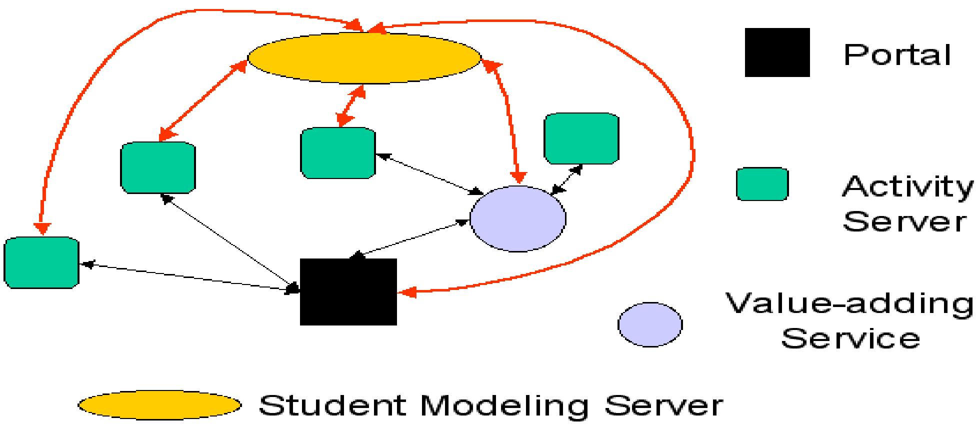


Figure 2.5: KnowledgeTree Architecture

Activity servers perform two roles in the KnowledgeTree system, the storage and delivery of learning activities. A course creator can query an activity server with their portal in order to locate relevant activities. A student can launch and interact with a remote activity using their portal. The student model server collects data about student performance and can be used by adaptive activity servers to provide personalised content delivery. Teachers are provided with the ability to create a course using a portal and multiple activity servers. The KnowledgeTree architecture is open and flexible allowing for the creation of multiple portals, activity servers and student modelling servers. Portals and activity servers communicate with each other using a standard communication protocol.

Research has taken place into the combination of APeLS and KnowledgeTree in order to provide a framework which combines the benefits of content reusability frameworks with the power of adaptive educational hypermedia systems [Brusilovsky et al., 2008]. The primary aim of the combination is the ability to reuse adaptive content independently of any specific AEH system. A number of key features were identified from each approach that should form the basis of an adaptive E-Learning solution. It was suggested that the course management system should be separated from the learning content. A portal should be provided to the end user that facilitates structured access to the educational content without the need to store it. The content can be retrieved from a number of content services which are independent of the portal and reside on a number of distributed servers. Portals can be maintained by course providers whilst content services can be maintained by content providers. Many portals can reuse the same content service thus allowing learning resources to be deployed in numerous different contexts. In contrast to content reusability frameworks, where a teacher would search for a learning object and integrate it into their course, it is suggested that a teacher should stop at the content specification stage. The portal should be capable of automatically finding or generating relevant learning content at runtime. This would solve the problem of outdated material appearing in courseware. Repositories can be constantly updated with new material and additional repositories can become available. If courseware was dynamically generated, the learner could benefit from the most up to date and appropriate content at all times, without a teacher having to re- author their course.

2.5 Interoperability

As E-Learning technology evolved it became more and more fragmented. New functionality was not always compatible with existing systems and content could not be shared amongst disparate environments. Several organisations emerged that aimed to develop technical standards, specifications and recommended best practice for E-Learning functionality. They consisted primarily of the IEEE Learning Technology Standards Committee (IEEE LTSC) [IEEE, 2007], the IMS Global Learning Consortium [IMS, 2010], Advanced Distributed Learning (ADL) [ADL, 2010] and the Aviation Industry CBT Committee (AICC)

[AICC, 2010]. These organisations have produced standards for learning content metadata, content packaging, pedagogy, user modelling and user testing. Metadata standardisation efforts have already been discussed in section 2.2. IEEE LOM [IEEE, 2002] and IMS LRM [IMS, 2001] were the primary standards to emerge for adding descriptive information to learning objects.

2.5.1 Packaging

The most prominent packaging approaches are IMS Content Packaging (IMS CP) [IMS, 2007] and the Shareable Content Object Reference Model (SCORM) [ADL, 2004]. IMS CP is designed to support the interoperability of content amongst various environments. The specification defines data structures that can be used to exchange data between systems that wish to import, export, aggregate and disaggregate packages of content [IMS, 2007]. Focus is placed on the packaging and transport of resources as opposed to the nature of those resources. This is because IMS CP supports content in a wide variety of formats and combinations.

The most widely used packaging approach in recent years is SCORM [ADL, 2004]. SCORM is, in fact, more than just a packaging approach as it contains a set of specifications concerning the development, packaging and delivery of learning objects. The primary aim of SCORM is courseware portability and interoperability. The SCORM 2004 specification consists of four books; The SCORM Overview, The SCORM Content and Aggregation Model (CAM), The SCORM Run-Time Environment (RTE), and the SCORM Sequencing and Navigation (SN). SCORM is built on top of specifications developed by other organisations. The SCORM CAM provides for the packaging and description of learning content. IMS CP is used to define the SCORM CAM and IEEE LOM can be used for descriptive metadata. The SCORM SN provides a mechanism for learning content to be sequenced into an order suitable for delivery to a learner. It relies on the IMS Simple Sequencing (SS) specification [IMS, 2003b]. Two sequencing mechanisms are supported, namely sequencing control modes and sequencing rules. Sequencing control modes involve the ordering of clusters of content that exist at the same aggregation level. Sequencing rules allow for a series of condition-action rules to be described by the course creator. The SCORM RTE provides an API that allows E-Learning systems to communicate with SCORM packages. The SCORM RTE relies on the IEEE ECMAScript Application Programming Interface for Content to Runtime Services Communications.

One of the most recent specifications capable of supporting content packaging and representing instructional designs is IMS Learning Design (IMS LD) [IMS, 2003a]. IMS LD "aims to represent the learning design of units of learning in a semantic, formal and machine interpretable way" [Koper et al., 2004]. It is based on a theatre script metaphor where a learning design consists of a set of plays which contain acts. Each act has a set of role parts that link specific activities to specific roles. The workflow is managed by the "method", which follows a set of prerequisites and learning objectives. IMS LD can support three levels of learning design complexity.

2.5.2 User Modelling

In section 2.3, the role of user models in the personalisation process of AEH systems was discussed. Two standards emerged in order to define user models in an interoperable manner, the IMS Learner Information Package (LIP) [IMS, 2005a] and IEEE Public and Private Information (PAPI) [IEEE, 1997]. The IMS LIP was designed to allow learner information to be shared between different systems. It consists of 11 categories; Identification, Goal, Qualifications, Activity, Transcript, Interest, Competency, Affiliation, Accessibility, SecurityKey, and Relationship. The Accessibility category can model disability preferences, physical preferences and technological preferences. In order to provide additional information on a users ability to interact with a system, an extension to LIP, IMS Accessibility for LIP (ACCLIP) [IMS, 2005b], was created. This specification is not restricted to issues of disability but also to scenarios where alternative modes of presentation may be required. The use of video captioning in a noisy environment or alternative control methods when a user is driving, are examples of these scenarios.

The IEEE PAPI [IEEE, 1997] was designed to allow the creation of student records which could be exchanged amongst educational systems. PAPI consists of four areas; Personal information, Preference information, Performance information and Portfolio information. Personal information consists of private data such as a name and address. Preference information can be publicly exchanged amongst systems and contains details such as learning style or disabilities. Performance information may include grades and data logs created by the learner as they interact with a system. Portfolio information consists of a learner's academic achievements. The standard permits different views of the information to be provided to different user types; teachers, parents, employers, etc.

2.5.3 Testing

As the testing of a students knowledge and understanding of course material is an important part of any E-Learning environment, a specification for interoperable tests was developed. The IMS Question and Test Interoperability (QTI) [QTI, 2006] specification provides a mechanism for teachers to define and reuse tests on various topics. Any QTI compatible environment is capable of administering a QTI test. The majority of implementations can not only score the tests but the results are also interoperable amongst various systems.

2.5.4 Data Abstraction

One final approach to interoperability must be mentioned. As E-Learning standards and specifications emerge and evolve constantly, there is a danger of restricting the capabilities of a system by tightly coupling it to a specific standard. In order to avoid this issue, some researchers are taking an approach where information is represented using abstract data models based on open description languages, such as XML (Extensible Markup Language)[W3C, 2008] [Dagger, 2006] or OCL (Object Constraint Language) [Warmer and Kleppe, 2003] [Melia, 2009]. Data can be modelled independently of any specific standard thus reducing any restrictions that a specific standard might enforce. In order to exchange information amongst numerous systems, the content in the abstract models can be translated into a format that suits the interoperable specification supported by the target system.

2.6 Summary

This chapter provided a high level overview of the areas of technology enhanced learning that are most relevant to the research in this thesis. Section 2.2 discussed the trend towards componentisation in mainstream E-Learning. Learning objects, learning object metadata and learning object repositories were outlined. Section 2.3 provided an introduction to Adaptive Educational Hypermedia Systems and discussed various axes of personalisation that can be used to deliver adaptive courseware. The section also discussed some model driven approaches that have been used to good effect in various AEH implementations. Section 2.4 outlined architecture standardisation efforts that have taken place in the domain of Technology Enhanced Learning (TEL). It also provided an overview of AEH systems that harness the capabilities of Service Oriented Architectures (SOA). Section 2.5 identified the primary organisations involved in standardisation efforts. Standards for content metadata, content packaging, pedagogy, user modelling and user testing were outlined.

Chapter 3

State Of The Art, Accessible Graphics

3.1 Introduction

This chapter discusses the state of the art in providing blind learners with access to graphical information. The chapter provides a background to the technologies and approaches for accessible graphics that relate to the research discussed in this thesis. Section 3.2 discusses the use of physical models in order to illustrate graphical concepts. Section 3.3 outlines the difficulties with designing accessible graphics for blind learners. A brief overview of standard guidelines is provided. A discussion of tactile production methods take place in section 3.4. Manual and digital methods of tactile authoring and creation are outlined. Techniques used by blind learners to explore tactile graphics are discussed in section 3.5. The cognitive difficulties experienced by learners attempting to interpret tactile graphics are outlined. The use of supplementary information to provide descriptive information relating to tactile graphics is illustrated in section 3.6. This is followed by a discussion of an approach that relies heavily on supplementary information in section 3.7. An overview of audio tactile graphics is provided along with details of specific systems. Section 3.8 provides a brief overview of additional interfaces that can be used for the presentation of and interaction with accessible graphics. The chapter ends in section 3.9 where specific issues that remain to be solved in the area of multimodal graphic creation are outlined.

3.2 Physical Models

If a graphical concept is being taught to a learner, it is advised, where possible, to provide the learner with a physical object that they can explore in a tactile fashion [Sheppard and Aldrich, 2001] [NCTD, 2010]. The learner can feel the physical object to gain an understanding of its shape and layout. For example, if the topic is the human skull, the learner should be handed a human skull to explore. Physically touching an object allows the learner to gain a real world understanding of a concept. When a real version of an object is not available, for example a fly, an elephant or a chemical, the use of physical 3D models is advised. Objects that could not otherwise be touched are scaled and rendered as a 3D model that the learner can explore.

The negative aspects to using 3D models are the availability and portability of three dimensional objects. It is generally not possible for the learner to take a 3D model home with them for independent learning. Additionally, mass producing physical models is time consuming and expensive. If a learner is operating in a distance learning scenario, a physical model is not an option. For these reasons, tactile graphics have become the most common method of providing graphical material to blind learners.

A tactile graphic is a raised image on a sheet of paper. A learner can explore the image with their finger in order to gain an understanding of the concept it represents [Edman, 1992]. Although tactile graphics cannot represent three dimensional concepts, they can adequately represent two dimensional alternatives, such as the cross section of the human skull from various angles. The lack of three dimensional representation is offset with the portability and mass production benefits of tactile graphics.

3.3 Tactile Design

A graphic suitable for delivery to sighted learners is not directly suitable for tactile representation. As the human eye can process data at a higher resolution than the human finger, graphics must be simplified before they can be delivered in a tactile form. The design of a usable tactile graphic requires the talents of a skilled tactile graphic creator.

Guidelines have been produced by international institutions such as The American

Printing House for the Blind [APH, 2010], The Canadian Braille Authority [CBA, 2010], The Braille Authority of North America [BANA, 2010] and many more. Designing a suitable tactile graphic requires significant skill and knowledge in order to simplify a graphic for tactile comprehension. The guidelines suggest that it may be hard for a learner to differentiate between symbols and lines placed closer than a quarter of an inch apart [APH, 1997]. Shapes with sides less than half an inch long may not be recognizable [CBA, 2003]. Lines, points and Braille should be separated by at least an eight of an inch, even if it introduces spatial distortion [BANA, 1983]. They are just a small number of the considerations that should be taken into account when designing a tactile graphic for a Blind learner. They clearly illustrate the complexity of designing a tactile graphic suitable for exploration.

3.4 Tactile Production

Various methods of tactile graphic production have evolved that range from highly skilled slow manual methods to simpler rapid digital methods [NCTD, 2010]. A brief overview of some of the available methods is provided in the following sections.

3.4.1 Manual

3.4.1.1 Craft

Craft, also known as collage or model making, involves the layering of material and objects on to a surface in order to produce a tactile representation of a graphic. A wide variety of items can be used to create the collage. Materials such as threads, wire, cloths of varying thickness and textures, corks, foam, wood and so on, can be combined to produce the affect desired by the tactile creator. Braille can be added by printing using an embosser and then cutting and pasting the labels onto the diagram. Advantages of the collage method are that the layering of objects can produce multiple tactile levels and provide the student with a sense of depth. Collage diagrams are also very durable and potentially cheap to make. Disadvantages of the method are that it can be quite time consuming to produce a diagram and difficult to mass produce in their original form.

3.4.1.2 Free-hand

Free-hand diagrams are generally hand drawn in real time using a variety of techniques. German film is a fine film that can be drawn upon with a small pointed tool. When pressure is applied the film creases and leaves a raised line for the learner to feel. Aluminium foil is a similar substance that can be drawn upon to form temporary graphics. Free-hand diagrams are easy to draw and are useful for graphics containing lines and basic shapes. They are also relatively cheap to produce. The downside is that they are not mass producible, typically not durable and it can be difficult to create complex graphics.

3.4.1.3 Thermoform

Thermoform, also known as vacuum form, is a process that is generally used for mass producing plastic copies of a master tactile graphic. A collage master is covered with plastic which is then heated and vacuumed. Once the plastic cools it is an exact replica of the master tactile and can be handed out to the learners. Benefits of this approach are that thermoform diagrams have good height and texture variations due to the detail of the original master. It is easy to produce multiple copies from a single master. Thermoform diagrams are durable as they can be wiped clean and they are cheap to make. There are negative aspects to this approach. The production of the initial collage master is labour intensive and requires a skilled creator. Thermoform machines are expensive costing up to 3000 sterling and provide a slow production speed. Graphics of this type cannot contain colour or textual information.

3.4.2 Digital

The manual approaches discussed above are quite time consuming and require a skilled tactile creator. The approaches in the following sections emerged in recent years in order to reduce the complexity of tactile graphic creation. Both approaches avail of computer software and can be used to mass produce tactile graphics at minimal cost and at a rapid pace.

3.4.2.1 Embossed

Embossed graphics, also known as Braille graphics, are paper graphics that are represented using Braille dots. The embosser punches dots into the paper and arranges them in such a way as to form a graphic. Embossed graphics can be created in various ways depending on the combination of software and hardware available. Standard Braille embossers can produce Braille graphics, however, specialist embossers are generally used to give better results. Advanced embossers, such as the Tiger [ViewPlus, 2009a], can print straight from Windows applications such as Excel and Word. Newer models, such as the Emprint Spotdot [ViewPlus, 2010b] can produce colour diagrams with a variation of dot heights to provide more detail, textures and layers. Once a graphic is available, all an author must do is print it making embossed diagrams easy to mass produce. Embossed graphics can often be limited in the number of shapes and infills they can reproduce. The majority do not print in colour and have no height variation in their dots.

3.4.2.2 Swell Paper

Swell paper, also known as microcapsule paper, is a type of paper that has microcapsules of alcohol embedded inside it. Graphics can be printed onto the paper and when it is exposed to a heat source the microcapsules that have been covered in black ink burst and swell to form a tactile graphic. Various methods can be used to place an image onto swell paper. An image can be photocopied onto the paper from another source or can be printed directly onto the page using an inkjet or laser printer. It is possible to draw directly onto the paper using black marker pens or a heat pen can be used. A heat pen is a pen with a heated tip that can be used to draw directly onto swell paper. The results can be instantly seen without the need to pass the paper through a heat fuser. As colour ink does not swell it can be used for labelling and to enhance the graphic for low vision learners. Swell paper graphics are easy to update and edit in bulk. The downsides of this approach are that black areas can smudge easily and leave ink on the learners fingers. There is little height variation and a limited number of useful fills and textures.

3.5 Tactile Exploration

Those who are sighted posses a huge library of images and their relationships to other images that reside in the brain [Herring, 2007]. Blind learners, particularly those who are congenitally blind have an absence of previous experience leading to no cognitive map [Parkes, 1994]. This often leads to an inability to grasp the bird's eye view perspective used in most graphical images. The difficulty in interpreting tactile graphics is often a disincentive to their use. A lot of blind learners do not possess the skills to explore tactile graphics in a coherent manner [Aldrich et al., 2002]. However, "this does not mean that useful spatial information cannot be obtained from tactile maps or that the congenitally blind should be denied the opportunity for using them" [Dodds, 1988].

Researchers have identified the importance of teaching children how to explore tactile graphics in order to form a tactile memory [Nolan and Morris, 1971]. The ability to understand tactile graphics has been termed 'graphicacy' by Aldrich and Sheppard [Aldrich and Sheppard, 2000]. Not only do learners need to build a cognitive map of tactile shapes but also develop suitable strategies for exploring tactile graphics. Various strategies for tactile exploration have been researched and defined [Cryer and Gunn, 2008]. Most learners will begin their exploration by scanning the entire graphic to gain a sense of the size of the graphic, how many elements it contains, and any distinctive features. Only after this initial scan will the learner begin to explore specific elements of the graphic. Their method of exploration is often systematic. An initial scan can be followed with an exploration of the edges before exploring the finer detail of each graphical element. Systematic exploration will often involve following a horizontal or vertical method of graphic exploration. Explorers may also perform line tracing and distinctive feature analysis of each element in a tactile graphic.

3.6 Supplementary Information

Regardless of the method of exploration used, a tactile graphic is of no use to a learner without some form of supplementary information [Kennedy et al., 1991]. A sighted learner can differentiate between visual concepts easily even if the concepts share similar characteristics. For example, a sighted learner can tell the difference between the different planets in a graphic depicting the solar system. A tactile of the solar system to a blind learner could feel like a number of circles with no explanation of what each circle represents or the relationship between them. Even though a sighted learner can visually identify differences between the planets they may not know what the planets are. For this reason graphics are generally labelled to provide additional information to the sighted learner. Without the provision of supplementary information a blind learner would interpret a diagram as a collection of meaningless shapes [Landau and Wells, 2003]. Blind learners "need very careful, time consuming explanation, they cannot just be presented with a diagram and understand" [Sheppard and Aldrich, 2001]. Therefore, information should be provided regarding the topic the graphic relates to, what is depicted on it, what each tactile element represents and additional information for each element where possible.

A number of approaches have been developed to provide supplementary information for tactile graphics. There is a perceived difficulty amongst instructors in relation to creating and delivering tactile graphics, therefore they often "try to avoid diagrams wherever possible and use description instead" [Sheppard and Aldrich, 2001]. When tactile graphics are used, they are often supplemented with verbal description on the part of the instructor. The instructor will verbally provide background information for a graphic and explain the various tactile elements within it. The need to verbally describe each graphic as a learner explores it is time consuming on the instructor. In addition, as an instructor is required to provide the supplementary information, the independent learning capabilities of the learner are reduced.

In order to combat this, some instructors began to record their verbal descriptions in an audio format (CD, Cassette, MP3, etc) and provide it to the learner along with the relevant tactile graphic [Sheppard and Aldrich, 2001]. The learner could play the audio description, follow the instructions provided by the instructor and receive supplementary information relating to the graphic. Although time consuming to create the initial audio, it removed the need for an instructor to verbally describe the graphic on each use. Additionally, the instructor could take the time to design a suitable audio guidance strategy to aid the learner in exploring the graphic in a suitable manner. The existence of the audio track allows a learner to operate independently without relying on a sighted guide. Issues arise however when a learners' library of tactile graphics begins to grow. The learner must be provided with a way of locating the correct audio description to match the graphic they wish to explore. This can become a complex and time consuming procedure as the tactile graphic and audio description are independent of one another and not linked in any way.

In order to provide a solution where the supplementary information is not decoupled from the tactile graphic, textual information can be provided on the graphic itself. One method of implementing such an approach is to provide a legend alongside the tactile graphic [Miele et al., 2006]. The legend contains a number of keys with associated values and is generally represented using Braille. For example, the key A1 might contain a value of "the moon". When a learner is exploring the tactile, the element representing the moon would have the key A1 printed on or beside it. The learner can either memorise the legend prior to exploring the tactile or move back and forth between the graphic and the legend as a new element is found. A procedure like this adds cognitive load to the learner during their exploration of the diagram, either through the need for memorisation or the requirement to move back and forth between the element in the graphic and its entry in the legend. It also requires a learner to possess a strong spatial awareness capability so as not to get disoriented when moving between the legend and the tactile graphic. In addition, a minimal amount of information can be provided as there is limited space for the legend data. In order to localise a tactile, a version must be produced with a legend for each language.

In order to reduce the complexity of the approach, the information for a given element can be placed directly on the graphic by means of Braille labelling. A label containing information relevant to a tactile element can be located beside the element. As a learner explores the graphic they can read the label beside each element to receive supplementary information relating to that element. This can reduce the confusion of moving back and forth to a legend, however the location of the labels has been known to cause disorientation to learners [Aldrich and Sheppard, 2001]. It is not always evident by their placement, which tactile element a label refers to. As space is limited Braille labels can only be used to provide a minimal amount of supplementary information. In order to localise a tactile graphic, a version must be produced with suitable labels in each language. The use of labels or a legend rely on the learner possessing a certain level of Braille literacy. As less than 10% of blind learners can read Braille, the methods above would be useless to those learners [Landau and Gourgey, 2001].

It can be seen that the approaches above contained useful solutions for providing supplementary information but they also introduced additional problems. An approach was required which could integrate the benefits of each approach into a single solution. The use of audio description would not require the learner to be Braille literate. Providing a link between the audio description and the tactile graphic would reduce the complexity on the learner to synchronise the correct content. Tools could be provided to aid in the creation of tactile graphics thus reducing the complexity on the instructor. The use of audio would solve the space restriction issue that exists with Braille labelling. Graphics could be localised by providing suitable audio descriptions in other languages. An approach emerged, known as audio tactile graphics, that harnessed the benefits outlined above.

3.7 Audio Tactiles

Audio tactiles are a combination of tactile graphics and a device capable of providing auditory feedback. The most common form of implementation involves a touch sensitive screen connected to a computer [Miele and Schaack, 2008]. The tactile graphic is placed on top of the touch sensitive screen by the learner when they wish to explore it. A learner interacting with an audio tactile is illustrated in figure 3.1. The tactile can be explored like a normal tactile graphic but in order to receive supplementary information the learner can press on specific regions of the tactile. When a region has been pressed, suitable supplementary information relating to that region can be provided to the learner. The information is usually provided in an auditory form, either using text to speech or recorded speech. Depending on implementation, other types of information can be provided such as non speech sound, access to documents or hyperlinks to websites. Most implementations also allow for multiple layers of information to be provided. On an initial press of a region, the label for that region will be spoken, for example “The Liver”. On each subsequent press of the same region an additional layer of information will be spoken to provide additional detail about the tactile element at that region. A generic architecture for an audio tactile system can be seen in figure 3.2. Audio tactiles provide access to information that would be impossible to incorporate into a diagram alone [NCTD, 2010]. This brings an entirely new multimodal dimension to the learning experience.

Audio tactile graphics were pioneered in the late 1980's by Donald Parkes of the University of New South Wales. When his colleague Reginald Golledge became blind, it became apparent to Parkes that the limitations of tactile graphics would make it almost impossible for Golledge to stay up to date in his field of Cartography [Pennisi, 1992]. He began to investigate ways to put tactile maps into an electronic format. The outcome of his research was the Nomad audio tactile system [Parkes, 1988]. Nomad consisted of an 18 by 15 inch touch sensitive screen that was fitted with a speech synthesizer. The screen was used in tandem with a computer containing the Nomad software. In order to interact with the device, the learner placed a tactile graphic on top of the screen and informed the computer what the graphic was. The learner could then press on areas of the graphic and the computer would speak relevant information. Pressing again at the same location caused the computer to provide more detailed information. Nomad also allowed learners to annotate audio tactiles with their own information. A facility was provided for the learner to type information into the computer that could be linked to a particular spot on the tactile.

Figure 3.1: Audio Tactile

3.7.1 Nomad Components

The Nomad system contained four components; the CAD system, Kernel, Information Access System and Walkabout System [Parkes, 1994]. The CAD system is a tool that enabled "totally blind and partially sighted people to create tactile and print drawings in a manner that is very similar to that which would be used by sighted persons" [Parkes, 1994]. A tactile template was provided containing a number of icons that "allow the blind user to select a range of Euclidean shapes, draw freehand and label the entities drawn" [Parkes, 1994]. The designs that were created using the CAD system could be embossed directly from the system or saved for input into supplementary software called Picture Braille [Pentronics, 2010]. The completed tactile could be placed onto the Nomad touchscreen and the labels would be spoken to the learner as they pressed on parts of the graphic. This process of tactile creation allowed blind users to visually represent their thoughts and thus interact with sighted colleagues.

The Nomad Kernel was the core of the system. The Kernel performed the task of linking the auditory information to the relevant graphic. Functionality was provided to set the scale of the graphic, calculate route distances and prepare and read line, bar and circle graphs. Text files could be scanned and placed at any position on the tactile graphic. It was also possible to paint with sound in nine different frequencies and draw and emboss graphics directly to an embosser. In total, 75 commands were provided to the user to control the Nomad Kernel.

The Information Access System was the primary Nomad utility and was used to allow the user to attach auditory information to graphics that could be used with the Nomad touchscreen. A tactile template was provided with the utility that contained a Braille alphabet along the top, a touch accessible user guide and four tactile function buttons. The first button read the title of the graphic that was placed on the touchscreen. The second measured linear distances between two points. The third provided more detailed information about any element on the graphic. The fourth button allowed the user to touch any three Braille characters that formed the start of the name of a feature in the graphic. Once pressed the system would direct the user to the feature using a series of "up, down, left, right" auditory instructions.

As all information could be provided in an auditory form, Braille literacy was not required. In addition, digitised sound allowed graphics to be prepared in any language. Nomad allowed audio tactile creators to mix speech with real world sounds, for example information could be provided about a waterfall followed by the sound of the waterfall. Nomad opened up independent learning capabilities to its users by providing the software for a given graphic on disk along with a tactile copy of the relevant image. If the learner had a version of Nomad at home, they could load the graphic onto their device and interact with it in an autonomous fashion.

The final component of Nomad was called Walkabout. The Walkabout system allowed a user to place a tactile map onto the Nomad touchscreen and trace a path along a proposed route that the user planned to make. The system would take note of all points of interest that lay adjacent to the route the user wished to travel. Walkabout then allowed the user to listen to the proposed journey in real time in order to preview it. The audio route could be played back on the computer that created it or saved to disk and played back on any audio capable computer. Placing the audio on a portable computer and receiving real time guidance was envisaged. Alternatively, the audio could be output to cassette and taken along to provide the user with guidance as they navigated the route.

Nomad was made available as a commercial device, formed the basis of numerous other commercial devices and was used as inspiration for various research projects [Loetzsch, 1994] [Loetzsch and Roedig, 1996] [Gardner and Bulatov, 2006] [Landau and Wells, 2003]. It did not however gain widespread appeal. The creation of the tactile graphics and the complimentary audio program could be a laborious process outside the capabilities of most users. Therefore, the technology was mainly used by large institutions to create special sets of audio tactile capable curricula [Gardner and Bulatov, 2006]. The touchscreen device the system was based on was expensive to purchase and contained a low tactile resolution. Although Nomad was not a commercial success, its design and techniques would be used as the basis for future audio tactile approaches.

3.7.2 Commercial Approaches

There are currently two commercially available touchscreen based audio tactile approaches, T3 [TouchGraphics, 2010a] developed by TouchGraphics [TouchGraphics, 2010b] and IVEO [ViewPlus, 2010a] developed by ViewPlus [ViewPlus, 2010c]. The approaches contain similarities but also differ in a number of key areas. Details are provided for each approach in a number of areas discussed in the following sections.

3.7.2.1 Equipment

Each approach utilizes a touch sensitive screen that is connected to a computer via a USB cable. The learner is required to place tactile graphics on top of the touchscreen and secure them in place. In the case of T3, a hinge can be raised to allow the tactile graphic to slide underneath. Once in place the hinge is lowered thus stopping the tactile graphic from moving [Landau and Gourgey, 2001]. IVEO's touchscreen contains a clamp at the top that should be lifted up to allow the tactile graphic to be positioned against the top edge of the touchscreen. In the portable version of the screen the clamp is on the left hand side. Once the tactile graphic is in place the clamp is lowered to secure the tactile graphic and prevent it from moving [Gardner et al., 2005]. Both touchscreens are single touch devices and in essence represent a large trackpad. Pressing on different areas of the touchscreen will move the onscreen mouse cursor to different areas of the computer desktop. The hardware being used is most commonly installed directly in front of a computers display in order to provide touchscreen interaction where users can activate on screen icons by pressing on them. In this instance the touchscreen is housed in a casing external to the computer [Landau and Gourgey, 2004]. A user can still interact with icons on the computers screen but as their are no images on the touch screen it is more difficult to synchronise screen presses with the onscreen cursor. It is the equivalent of separating the touchscreen element of a smartphone from its display and attempting to select an icon. Because of this separation, each approach requires the user to calibrate their touchscreen on each use in order to ensure that the resolution of the touchscreen and the resolution of the visual display are synchronised. If they are not, then a learner may be provided with inaccurate audio feedback when pressing on regions of the tactile graphic.

3.7.2.2 Creation Methodology

One of the most significant variations between the two approaches is the method of audio tactile creation provided to the end user. T3 opts for a multi phase process where the touchscreen takes a prominent role in the creation of an audio tactile [Rosenblum et al., 2004]. The creation software is known as the "TTT Authoring Tool". Each tactile graphic used with T3 contains a tactile user interface. The authoring tool contains an onscreen representation of the T3 tactile interface. Previous iterations of the tool provided creators with an image file that represented an empty T3 tactile interface. The creator could use a drawing package in order to fill in the title, plate number and ID code for their tactile. They could also place a suitable tactile image inside the workspace of the template. When all steps had been completed the entire template was printed using their production method of choice, most commonly swell paper. In the most recent iteration of T3, template sheets are provided to creators prior to audio tactile creation. A template sheet is a vacuum formed representation of an empty T3 tactile interface. Additional template sheets can be purchased from Touch Graphics in bundles of 25. Creators can use any method of tactile graphic design and production they desire as long as they place the final tactile graphic inside the workspace of the T3 template. The workspace dimensions are nine by eleven inches and the graphics should be produced in order to fit in the available space.

In order to attach supplementary information to the tactile graphic, the authoring tool must be made aware of the location of the tactile elements in the template sheets workspace. We will refer to this as annotation. In order to do so the creator informs the authoring tool of the type of annotation they wish to create. Three types of annotation shape are supported; Point, Line and Region. A point is a single point on the tactile, a line can be a two point line or a multipoint line and a region is a multi point area with a border like a polygon. Once a type of annotation has been chosen, the creator presses on a specific area of the tactile graphic in order to create an annotation for the area. A visual representation is created in the authoring tool to represent the area the creator has touched. For example, if there is a tactile square on the graphic, the creator would select the "Region" type, and press on the four corners of the square. Once completed the authoring tool would show a square annotation at the same location in the onscreen template as the printed template. Once an annotation is in place, supplementary information can be provided for it. Each annotation can contain a title and various additional layers of information. This information can be provided by typing it into a textbox, recording it using built in recording functionality, importing a sound file, or importing a text file with the information. This process is repeated in order to annotate each element of the tactile graphic. Background information can also be provided for a tactile using the same methods.

The IVEO audio tactile creation methodology does not require the use of it's touch-screen. All of the creation functionality can be performed directly inside the IVEO Creator software [ViewPlus, 2009b]. The creator can author the tactile image in the drawing package of their choice. The IVEO system operates using the Scalable Vector Graphics (SVG) [W3C, 2010] format and therefore a creators authored image should be saved in the SVG format [Gardner et al., 2008]. Once a graphic is available it can be imported directly into the IVEO Creator system. Once imported the image appears on screen. The creator does not have to perform an additional annotation step as the SVG format provides the creator with the ability to immediately click on elements of the image. SVG images can therefore be considered as "pre-annotated". Functionality is provided in order to add supplementary information to each element on the image. Unlike T3, multiple independent layers of information are not possible. Creators can add a title and a description to an element. The description can contain more than one line of information but the learner will not be able to trigger each line one at a time in sequential order. Instead, the entire description will be read to the learner when they press on that element of the tactile. Creators can type a description using text, they can record the information using the built in recording functionality, they can link to a pre-existing audio file, they can link to a document or they can link to a webpage. This provides more variation in the types of supplementary information that can be provided to a learner compared to T3, but less fine grained control over the delivery of that information. The ability to add supplementary information without the annotation step reduces the complexity and the time consuming nature of the process for the creator. Background information can be provided in a similar manner to the T3 system.

IVEO provides some additional functionality for audio tactile creators. Firstly, it provides a converter tool that allows for graphics created in any Windows program to be exported in a suitable SVG format and imported into IVEO creator. In order to do so the creator draws the graphic in their tool of choice and clicks on print. IVEO converter appears in the print options and allows the user to create an SVG version of their graphic. The user can then import that graphic into IVEO Creator. It is also possible to import PDF files or scan documents directly into IVEO Creator. Creator contains advanced Optical Character Recognition technology that can digitise the document in a suitable format for IVEO to interact with. Finally, Creator contains some basic SVG drawing functionality that allows the user to draw a graphic directly into the Creator interface. Tools are provided to create Rectangles, Ovals, Polylines, Filled Polylines, Polygons and Curves. Titles and descriptions can then be added to the drawn image for delivery to the learner.

3.7.2.3 Data Model and Packaging

The T3 user interfaces were developed using Macromedia Director [Macromedia, 2010]. When a user executes a T3 interface they are actually loading a Macromedia Director project using the Macromedia Projector runtime. Each audio tactile that is produced is a Macromedia Director file. The tactile interface, annotated regions, layers of supplementary information and background details are all represented inside the Director data model. This data model is a proprietary standard used by Macromedia. Director projects are made up of numerous cast and sprite elements that are linked together using scenes. A Director project is similar to a DVD chapter menu where the interface jumps to specific scenes based on the selections a user has made. The T3 authoring tool is therefore an audio tactile specific interface sitting on top of a Macromedia Director backend. Everything that is needed for delivery of the audio tactile including any relevant audio files are packaged in the Director file.

IVEO relies on the SVG [W3C, 2010] graphic standard as a data model for its audio tactiles. SVG is an XML based standard and it's shape elements can contain unique identifiers. These identifiers can be used to trigger the playback of relevant supplementary content to the learner. The available shapes and their corresponding tags in the SVG standards are; Rectangle `<rect>`, Circle `<circle>`, Ellipse `<ellipse>`, Line `<line>`, Polyline `<polyline>`, Polygon `<polygon>` and Path `<path>`. Each shape tag can contain the child tags, `<title>`, `<desc>` and `<a>` for each shape. The IVEO Creator interface uses these tags to represent the title and supplementary information provided by an audio tactile creator for a given element. IVEO also uses it's own namespace inside it's SVG graphics for a number of tags relating to IVEO specific preferences and settings. All of the required data for an IVEO audio tactile is embedded inside the SVG graphical file. The only deviation from this is when a creator has chosen not to provide textual information and has linked to an audio file, document or webpage. In this case the `<a>` tag for a given element may contain a URI to the relevant file. If the file is being provided with the audio tactile there will be an accompanying folder alongside the SVG file containing the relevant files. A benefit of IVEO's use of SVG as a data model is that it allows for scalable graphics. Learners can zoom in and interact with a specific area of a tactile without compromising the detail of the image or the ability to interact with the audio tactile. Due to the static nature of the tactile, if a learner wishes to interact with a zoomed in version of an image they must also print a tactile with the same zoom level.

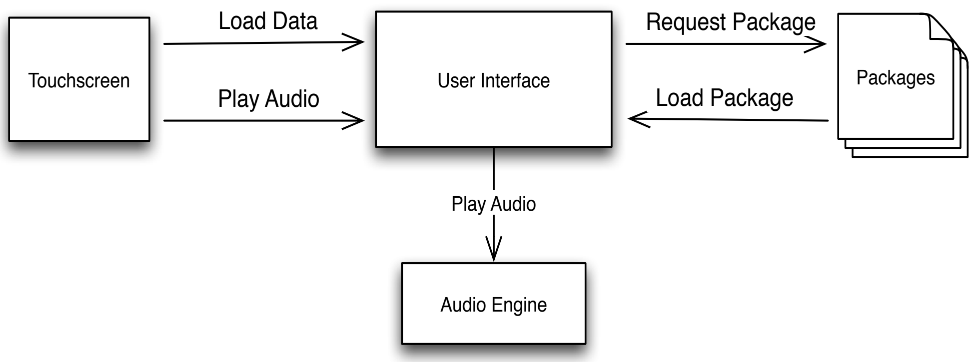


Figure 3.2: Generic Audio Tactile Architecture

3.8 Additional Interfaces

A number of other accessible diagram interfaces have been designed that either do not use tactile graphics or do not use touchscreen based interaction. These interfaces have been confined mostly to research projects but may emerge in the coming years as the standard methods of presenting and interacting with accessible graphical data. Although our work is focussed primarily on touchscreen based audio tactile graphics, it is designed to support other forms of graphical content presentation and interaction. Therefore, the projects that follow are provided as examples of where our solution may be of use as part of future work.

3.8.1 Digital Pen

A pen based audio tactile interface has been designed by Joshua Miele in the Smith Kettlewell Eye Research Institute. Miele identified that although touchscreen based audio tactiles were a significant innovation that addressed many of the problems of purely tactile diagrams, they suffered from shortcomings of their own. These shortcomings included a lack of portability, the need to frequently recalibrate the devices, providing a mechanism for the computer to identify the graphic, errors that occur when more than one finger is placed on a screen at one time and a high cost of ownership [Miele and Schaack, 2008].

A solution to these problems was found with the use of the Livescribe smartpen [LiveScribe, 2010] [Miele, 2007], a mainstream portable digital pen containing an internal computer. The pen is able to identify a specific graphic as well as its position on that graphic, circumventing the need to identify or calibrate the device as required by touchscreen based audio tactile systems. The pen contains a built in speaker and a headphone jack so that a computer is not required in order to provide auditory feedback. As no touchscreen is required, the user can explore the graphic with two hands without impacting on the accuracy of the pens feedback. The technology inside the pen can be seen in figure 3.3

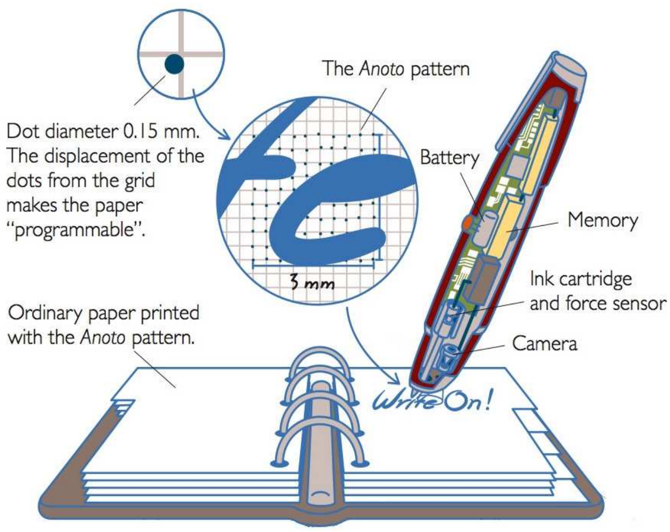


Figure 3.3: Livescribe Pen

In order for the approach to work, a microdot pattern must be printed onto the tactile graphic. This dot pattern is based on commercial technology from a Swedish company called Anoto [Anoto, 2010]. The dot pattern is printed onto the paper prior to embossing or vacuum forming. The pattern consists of a grid with a spacing of 0.3mm. A dot can be printed near the intersection of the grid offset at four possible positions. The pen contains a camera capable of identifying the variations in the dot patterns. It typically views a six by six group of dots at one time. Depending on the location of the dots that are currently visible to the camera, the pen can determine it's exact position on a page. This position is used to play relevant audio feedback to the user. A sample dot pattern can be seen in figure 3.4.

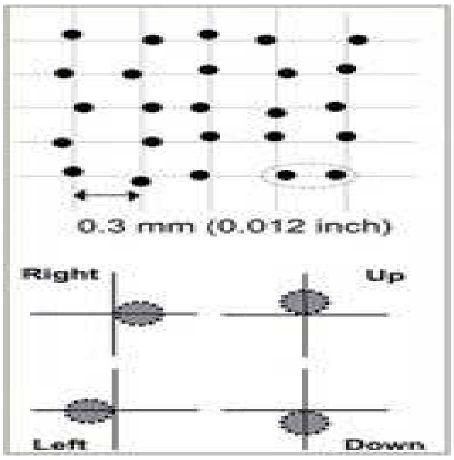


Figure 3.4: Livescribe Dots

In order to interact with a tactile graphic using the pen, the user explores the graphic with their fingers. When they reach an area they wish to receive information on they press the tip of the pen against it [Miele and Landau, 2010]. The pressure sensor in the pen triggers the camera to identify the location the pen has pressed and in turn select the appropriate auditory information [Landau et al., 2008]. Once located the information is played back using the pens speakers or through attached headphones. There is no internal text to speech engine in the pen, therefore all information is pre-recorded and stored inside the pen.

At this time a tool for end user authoring of suitable pen based audio tactile content is not available. Suitable material is being produced by TouchGraphics [TouchGraphics, 2010b] and materials are being released to the market in an incremental fashion. At time of writing, materials are to be released at six month intervals until the end of November 2011. In contrast to the methods of authoring and delivery used with touchscreen based systems, there is no single delivery program that can be used to open content stored in a particular data model or package structure. Each pen based audio tactile requires its own JAVA [JAVA, 2010] program in order to function. The program contains a digital copy of the relevant image, all audio information, and instructions on how to link them together. Once programmed, special software is required to sync the program into the pen. Although the interface provides benefits for the user it is not yet in a position to be used as a tool for content creators.

3.8.2 Haptic Interfaces

Researchers have experimented with entirely virtual representations of graphics. This type of interaction involves providing a learner with the sensation of force and texture required to illustrate a visual concept. The approach is known as haptics and is a form of kinaesthetic feedback. Special haptic devices, containing motors and servos are used to provide the learner with a sense of shape, density and texture through the varied application of force to the learners hand.

Various projects have explored haptics for the delivery of graphics to blind users. One such project is GRAB (Computer Graphic Access for Blind People) [GRAB, 2010], which developed a Haptic Audio Virtual Environment (HAVE). This environment provides users with the ability to locate and interact with three dimensional computer generated objects using the modalities of touch and audio [Wood et al., 2003]. A haptic interface was custom made for the project that allows the user to interact with the HAVE using two fingers. It consisted of two arms each capable of six degrees of freedom (DOF), three of which relate to finger position and the other three relate to finger orientation. A thimble was located at the end of each arm to allow the user to insert their fingers and control the device. The interface provided a large workspace that is 600mm wide, 400mm high and 400mm deep. A user can be seen interacting with the device in figure 3.5. The mechanics of the interface could be used to provide various types of functionality to the user. It could allow them to feel the shape of a virtual object and explore it's features, such as stiffness, softness, texture and weight. The users movement could be constrained by forcing them to follow specific trajectories or restricting them to the boundary of an object. Objects could be picked up and moved and users were free to zoom and pan the virtual workspace [Avizzano et al., 2003].

The core of the system was the Haptic Geometric Modeller (HGM), which allowed the user to interact with three dimensional virtual objects using haptic stimuli, sound aids and speech recognition [GRAB, 2010]. Users could control the interface via speech input and auditory feedback could be provided in combination with the haptic feedback. The HGM consisted of a C++ toolkit containing all the necessary algorithms for the haptic interaction, speech recognition and audio output. It was not directly coupled to the GRAB interface and therefore could be used with additional haptic interfaces.

Figure 3.5: GRAB User Interaction

The GRAB project has been used to allow blind users to view chart data, explore maps and engage with haptic games. Evaluations of the approach found that users appreciated the ability to simultaneously use two fingers in a single workspace, a facility lacking in other haptic interfaces. The workspace is also larger than that provided by other devices. Users found that the interface moved smoothly, was robust and contained a high level of positional accuracy. Users also appreciated the ability to use the modality of audio for both input and output [Avizzano et al., 2003].

Although GRAB developed it's own haptic interface, other projects, such as MICOLE (Multimodal Collaboration Environment for Inclusion of Visually Impaired Children) [MICOLE, 2010] [Pietrzak et al., 2007] provide blind users with access to virtual objects using commercial devices. These devices include the Sensable Phantom Omni [Sensable, 2010b], Phantom Desktop [Sensable, 2010a], Phantom Premium [Sensable, 2010c] and the Novint Falcon [Novint, 2010].

3.8.3 Refreshable Displays

Large refreshable Braille displays are an ideal way of delivering tactile graphics in a dynamic fashion. These devices provide Blind users with a similar interface to embossed graphics but in real time. One example of this type of interface can be seen in the HyperBraille project [HyperBraille, 2010]. The project has developed the BrailleDis 9000, a piezo-electric tactile graphic display consisting of 120 times 60 pins [Völkel et al., 2008], which can be seen in figure 3.6. The pin matrix is touch sensitive and is capable of interpreting multiple points of contact at once, thereby providing it with the multitouch capabilities that are becoming commonplace in mainstream human computer interaction. One of the most innovative elements of the BrailleDis is it's use of vertical Braille modules. Each module provides 2 times 5 pins and is connected with it's own rate sensor, actuator and data bus connection. This design allows for displays of different sizes to be constructed and for damaged modules to be easily replaced.

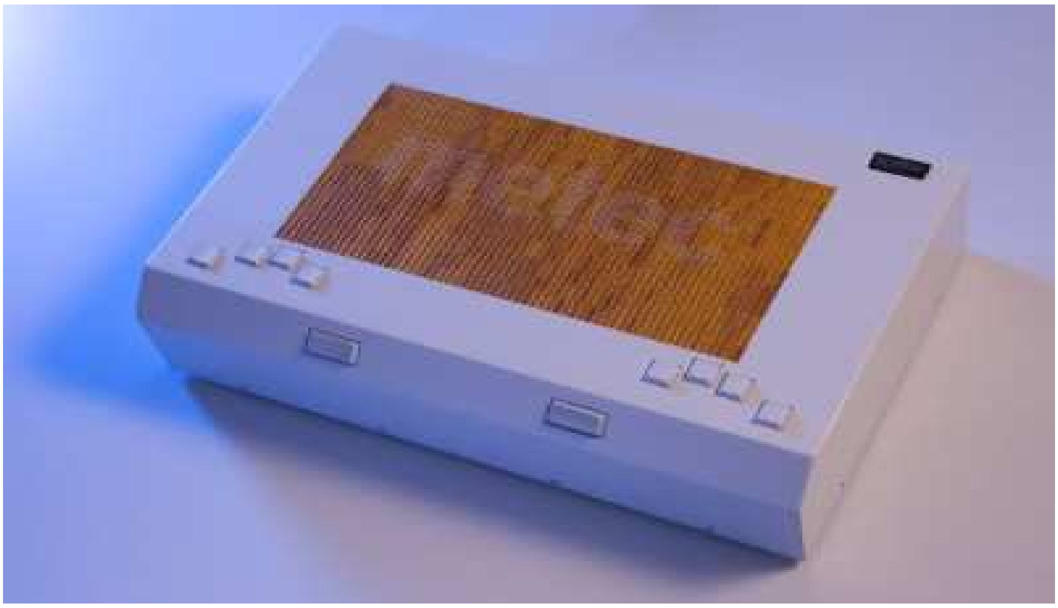


Figure 3.6: BrailleDis 9000

The device can be used to allow a user to interact with graphical concepts via touch and be provided with audio haptic feedback in a similar manner to audio tactile systems

[Weber, 2010]. This capability is supported by means of an off screen model (OSM) [Kraus et al., 2008]. The off screen model allows the system to compute the location on the display that the user has pressed. This location can be used to deliver appropriate audio or haptic feedback. Multitouch gestures [Schmidt and Weber, 2009] can also be used to facilitate panning and zooming of images, which compares with the capabilities of IVEO. Unlike IVEO however, where a tactile printout is required for each new level of zoom, the image on the BrailleDis updates dynamically. The system is currently being used to deliver OpenStreetMap GIS data to blind users using a combination of haptic and audio feedback [Zeng and Weber, 2010].

3.9 Remaining Issues

A lot of research and commercial work has been done in the area of tactile graphics. The majority of the work to date has focussed on new interfaces or new forms of graphical interaction. Content management and end user content creation is generally an afterthought as the majority of the work has focussed on the presentation of and interaction with graphical data. Because of this fact, tactile graphics remain predominantly difficult to make, involving a time consuming process best undertaken by skilled tactile graphic designers. Teachers are so concerned with the labour intensive nature of tactile production and support that they try to avoid using graphics wherever possible. This can fundamentally alter the learning experience and restrict independent learning possibilities. Audio tactile graphics go some way towards providing an acceptable method of tactile creation for non-experts but issues remain.

As T3 and IVEO are the only commercial audio tactile approaches with widespread availability for the end user, and as they are the most related work to this research the discussions will focus on them. The audio tactile approaches outlined contain benefits for the learner and have solved a lot of the highlighted problems in regard to tactile graphics, namely the use of Braille labelling and decoupled audio feedback. Some issues, that predominantly affect the creator, remain that require additional research. These issues can be categorised into reusability, producer support and interoperability. Each issue is discussed in the following sections and summarised in table 3.1.

3.9.1 Reusability

None of the approaches to date explicitly take into account the possibility that a creator may want to reuse individual elements of their audio tactile graphics. Imagine a scenario where a creator would like to alter the supplementary information for a given audio tactile but reuse the same image. As T3 uses the Macromedia Director data model, all of the supplementary information is tightly coupled to the image in the Director file. In order to alter the supplementary information the creator must open their audio tactile, delete all of the existing supplementary information, replace it with new information and save the graphic as a new Director file. Alternatively, they can perform the audio tactile authoring process from the beginning by importing and annotating the same image. IVEO relies on SVG as it's audio tactile data model. It exploits the `<title>`, `<desc>` and `<a>` tags of SVG to represent it's supplementary information. This also provides a tightly coupled connection between the image and the supplementary information. In order to make changes the creator must open their IVEO SVG file, delete all of the titles and descriptions from each graphical element and replace them with new content. The tactile is then saved as a new SVG file. Alternatively, the process for creating a new audio tactile can be performed from the beginning. In each approach unnecessary replication is taking place. There are now multiple copies of the same graphical image in order to provide different supplementary information. This is not efficient in terms of cost or time.

As learners can have different preferences for the production methods and textures used in their tactile graphics, a scenario may arise where a creator wishes to alter an image but reuse the same content. For example, imagine an audio tactile already exists for a map of Germany. A creator may find another tactile image containing more suitable textures for printing on an Embosser. It should be possible to reuse all of the supplementary information from the first tactile with the new image. Currently a creator would have to create the new tactile entirely from the beginning as the supplementary information is coupled to the original image.

One possibility does exist to provide reusable supplementary content. If a creator has been using external files for supplementary information these files are reusable outside of the audio tactile package of a given approach. T3 allows a creator to use audio files and text files, IVEO allows audio files, documents and hyperlinks. Even though it is technically possible to reuse those external files, no explicit functionality is provided to do so. Additionally, a certain level of replication must still take place as the external files are generally included with each audio tactile package.

Packaging is an acceptable method for supporting the exchange and delivery of content. However, when it is used for creation it causes a scenario where the individual components of an audio tactile are not independently reusable. Content replication is taking place each time a graphical image or piece of supplementary content is being reused. Content reusability is common in areas such as computer programming, Web Services and Technology Enhanced Learning. A reusable approach to audio tactile content could provide savings in time and cost that would be of benefit to audio tactile creators.

3.9.2 Producer Support

Although each approach provides a user manual and integrated help functionality to guide a creator through the use of their interface, there is limited support available for the location or creation of raw content. In order to create an audio tactile with current approaches, a certain level of tactile skill is required. The creator must either be capable of designing a suitable tactile image themselves or liaising with a tactile designer to inform them of the image they require. Some tactile repositories have begun to appear but they contain limited amounts of information and rarely provide supplementary information beyond the inclusion of Braille labels in the image [PRCVI, 2010] [TactileLibrary, 2010]. Therefore, there is very little support, especially for non-expert creators, in locating suitable content that can be used in the construction of their audio tactile. A mainstream teacher is not required to draw a graphic themselves, they can search the web for a suitable image and immediately reuse it. It is the authors view that a similar approach should be available for accessible graphic creation. An approach should be available that allows a creator to search for, retrieve and reuse individual graphical components that are useful for the topic they are creating an audio tactile for.

3.9.3 Interoperability

As previously discussed T3 and IVEO utilise different data model and packaging approaches. T3 is based on Macromedia Director and IVEO on SVG. Because of this the audio tactiles produced on one system cannot be used on the other. Conflicting data models and packaging approaches provide no interoperability between the two systems. In order for a creator to provide an audio tactile that can be used on either system they must author a version of their audio tactile on each system. This increases the amount of time and effort required for audio tactile creation. The ability to create an audio tactile once and have it delivered on any audio tactile platform would provide the learner with the ability to use their preferred interface while reducing the burden on the creator.

Features of Related Work		
Features	T3	IVEO
E-Learning Integration	✗	✗
Standards Based	✗	✓
Touchscreen Authoring	✓	✗
Reusability	✗	✗
Interoperability	✗	✗
Producer Support	✗	✗
Extensibility	✗	✗

Table 3.1: Related Work

3.10 Summary

This chapter discussed the state of the art in providing blind learners with access to graphical information. Section 3.2 discussed the use of physical models in order to illustrate graphical concepts. Section 3.3 outlined the difficulty with designing accessible graphics for blind learners. A brief overview of standard guidelines was provided. A discussion of tactile production methods took place in section 3.4. Manual and digital methods of tactile authoring and creation were outlined. Techniques used by blind learners to explore tactile graphics were discussed in section 3.5. The cognitive difficulties experienced by learners attempting to interpret tactile graphics were outlined. The use of supplementary information to provide descriptive information relating to tactile graphics was illustrated in section 3.6. This was followed by the discussion of an approach that relies heavily on supplementary information in section 3.7. An overview of audio tactile graphics was provided along with details of specific systems. Section 3.8 provided a brief overview of additional interfaces that can be used for the presentation of and interaction with accessible graphics. The chapter ended in section 3.9 where specific issues that remain to be solved in the area of multimodal graphic creation were outlined.

Chapter 4

Design

4.1 Introduction

This chapter discusses the design of a new approach for multimodal graphic assembly, which is the primary contribution of this research. The first research objective, as outlined in section 1.4, was to "Research and identify specific techniques in the areas of componentisation, information architecture and system architecture, that can be applied in the area of multimodal graphic assembly.". Chapter 2 provided an overview of some of the techniques from Technology Enhanced Learning (TEL), which relate to the categories defined in the first research objective. This chapter illustrates how elements of those techniques informed the design of the framework presented in this thesis.

During the initial period of research, work took place on the Audio Haptics for Visually Impaired Training and Education at a Distance (AHVITED) project [AHVITED, 2009]. AHVITED was a European Union funded research project that aimed to investigate the use of audio tactile graphics in distance learning environments. As the project was focussed primarily on the learner, it's aims and objectives are outside the scope of this thesis but can be found in the following publications [McMullen, 2008] [Fitzpatrick and McMullen, 2008] [McMullen and Fitzpatrick, 2008] [McMullen and Fitzpatrick, 2009]. As the project provided insights and empirical evidence into the issues faced by audio tactile creators, some of those insights informed the design of the framework in this research. Therefore, references will be made to AHVITED in this chapter, where appropriate.

The approach suggested by this research is called the Multimodal Graphic Assembly and Delivery Framework (MGADF) [McMullen, 2010]. Section 4.2 discusses the application of mainstream E-Learning architectural approaches in the design of the MGADF system architecture. Section 4.3 outlines the information architecture of the MGADF. The data model for each component is described and examples provided. Finally, section 4.4 describes methodologies that can be followed by non-expert multimodal graphic assemblers in order to create accessible graphics.

4.2 System Architecture

Siqueira stated that "e-learning initiatives should consider an e-learning infrastructure, which should be based on an e-learning architecture, which, in turn should be based on a generic approach" [Siqueira et al., 2008]. As our research relates to the integration of multimodal graphics in E-Learning environments, it can be described as an "e-learning initiative". Therefore, the system architecture of the MGADF is based on the generic architecture identified by Siqueira, shown previously in figure 2.4. "Each instance of the generic architecture can use a different set of components that can by themselves incorporate different learning approaches" [Siqueira et al., 2008]. As the aim of the MGADF is to support the assembly and delivery of multimodal graphics, the focus of the system architecture is on the "Content Selection", "Content Development" and "Visualization" components of the generic architecture.

As the generic architecture does not suggest any particular techniques for component design, specific approaches from Technology Enhanced Learning will be used to inform the design of the MGADF. In chapter 2, content reusability frameworks were discussed that allow course creators to search for, retrieve and sequence existing content in order to produce coherent courseware. As the course creator can work at a higher level of abstraction, savings can be made in terms of time and cost, which can lead to a simpler courseware construction procedure. The primary arguments made by instructors of blind learners against the use of accessible graphics relate to their production time and the complex nature of their creation [Sheppard and Aldrich, 2001]. Therefore, the MGADF will apply the techniques of content reusability frameworks to facilitate the assembly of multimodal graphics.

Dagger stated that "tight dependence restricts reusability, whereas loose abstraction facilitates reusability" [Dagger, 2006]. In chapter 3, we discussed related systems, IVEO [ViewPlus, 2010a] and T3 [TouchGraphics, 2010a], whose components were tightly coupled due to the design of their data models and packaging approaches. In order to decouple the components of a multimodal graphic for enhanced reusability, the relevant independent components must be identified as indicated in the second research objective, "Identify the independent components that form a multimodal graphic". Through investigation, the related systems were found to consist primarily of three core components; a visual component, numerous content components and a metadata component. The visual component relates to the image that appears on screen and on the printed tactile. Content components contain the supplementary information that is delivered to the learner as they interact with the visual component. They generally consist of a title and numerous layers of information. Depending on implementation, various types of layer content may exist, such as textual, auditory or hyperlink. The metadata component represents descriptive information relating to the multimodal graphic, such as a title or background description. In previous approaches, all three components were tightly coupled and not independently reusable. In the MGADF, the components will be decoupled from one another with the provision that they are capable of being combined into a coherent multimodal graphic. It should be noted that a multimodal graphic is deemed to be coherent if the content components that are played back to the learner match the visual component with which the learner is interacting. For example, a learner should not receive information about the human eye if they are exploring the human digestive system.

This leads us to the third research objective, "Design an extensible architecture capable of supporting the independent components of a multimodal graphic and allowing for their interaction". Recent generations of learning technology systems have embraced Service Oriented Architectures (SOA) as a means of supporting content separation and interaction [Dagger, 2006]. Primary examples of such approaches are the APeLS [Conlan and Wade, 2004] and KnowledgeTree [Brusilovsky, 2004b] systems discussed in chapter 2. Both approaches separate the primary components of their systems into independent data models that can be exposed using Web Services. The use of Web Services provides interoperability between components and the systems that communicate with them as they are based on open data standards and communication protocols.

APeLS and KnowledgeTree contain a number of key features that they suggest should be incorporated into the design of modern E-Learning approaches. Firstly, the interface of the E-Learning system should be separated from the content. This can be achieved by providing lightweight portals that communicate with content servers via Web Services. Secondly, content should be separated into numerous models and stored in repositories so that they are independently reusable.

The separation of system components into various models highlights the existence of multiple roles in the courseware construction procedure. Adaptive Educational Hypermedia (AEH) systems in particular, benefit from the contributions of individuals with different skillsets [Brady et al., 2005]. During the period of formative research, the existence of multiple roles in the multimodal graphic creation process was identified through the creation of pilot materials [McMullen and Fitzpatrick, 2010a]. The creation of the visual and content components and the assembly of multimodal graphics took place in different institutions. Project partners indicated that they would have appreciated the ability to operate independently on individual components and subsequently make those components available for others to use.

Both APeLS and KnowledgeTree were designed to be extensible. As they are based on Web Services and all interaction takes place using standard communication protocols, new services can be inserted into each approach in order to provide additional functionality and new instances of existing services can be created containing additional content. For example, new repositories of content can be created and immediately exposed to course creators via their portals.

As a consequence of the techniques discussed above, the system architecture of the MGADF is designed as a component based Service Oriented Architecture. The use of Web Services facilitates creator collaboration, content interoperability and extensibility. The architecture comprises a set of data integration services independent of any particular multimodal graphic assembly process. All assembly and display logic remains under the control of those developing interfaces for use with the framework. This allows for the content to be adapted for use with a variety of interfaces and interaction techniques. The primary aim of the architecture is to allow for each component to be stored, searched for and retrieved independently of one another whilst also allowing for their interaction. Each component is outlined in the following sections.

4.2.1 Services

The Identifier Service can be seen in figure 4.1 labelled with an "A". It provides unique identifiers that can be attached to each component in the framework. Identifiers are required in order to search for and retrieve an individual component for editing or delivery. An identifier repository is used in order to ensure that duplicate identifiers are never issued.

The Visual Object and Content Object Services are labelled "B" and "C" in figure 4.1. The services provide identical functionality but differ in the types of content they support. The role of each service is to provide functionality for the storage, search and retrieval of their designated components, either Visual Objects or Content Objects. Visual Objects represent the image the multimodal graphic is based on and Content Objects contain the supplementary information that can be provided to the learner. Each service is connected to a repository, which provides the persistent storage capabilities. Each service requests a unique identifier for every component from an Identifier Service before storage. The identifier can be used later to facilitate the search and retrieval capabilities of the service.

The Assembly File Service can be seen in figure 4.1 labelled with a "D". The role of this service is to provide functionality for the storage, search and retrieval of Assembly Files. Assembly Files contain the metadata for a completed multimodal graphic and the location and identifiers of the components that are required to reconstruct it. The service is connected to an Assembly File Repository, which provides the persistent storage capabilities. The Assembly File Service requests a unique identifier for each Assembly File from an Identifier Service before storage. The identifier can be used later to facilitate the search and retrieval capabilities of the service.

4.2.2 Assembly Architecture

The entire architecture can be seen in figure 4.1. The architecture was required to be bi-directional allowing for the assembly and delivery of multimodal graphics. In figure 4.1 an Assembly Interface is visible interacting with the architecture. The logic required to decompose a graphic into it's component parts for storage and compose independent components into a coherent multimodal graphic can be seen in the interface. These functionalities are referred to as the Decomposition Engine and the Composition Engine. It is up to each individual interface developer to design and implement each element of functionality to suit their own purpose. The architecture allows for the Assembly Interface to send Visual Objects, Content Objects and Assembly Files through the services for storage in the relevant repositories. It is equally capable of operating in reverse and allowing an Assembly Interface to search for and retrieve components from the repositories using the services. The reason for retrieval in an Assembly Interface is to allow an assembler to edit a multimodal graphic after it has been created and to support non-expert multimodal graphic assemblers in searching for suitable content to reuse.

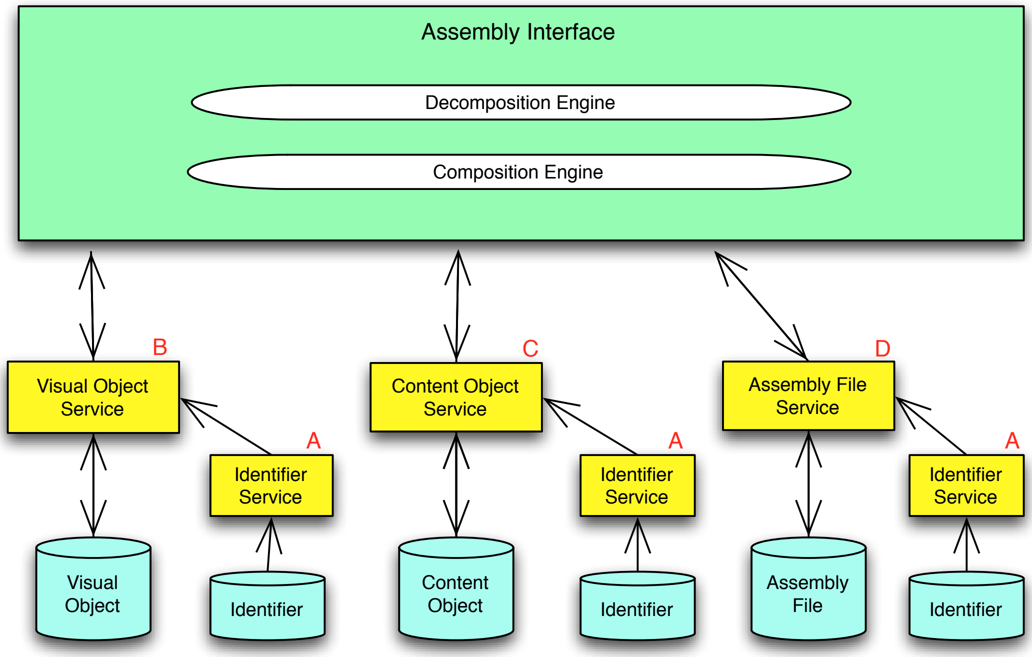


Figure 4.1: Assembly Architecture

As each component is represented with its own service, multiple independent roles can be supported. For example, Visual Component Creators can be provided with an additional interface whose only function is to provide access to a Visual Object Service. The creator can use the facilities of the service to add, search for and retrieve Visual Objects. The Visual Component Creator would not be required to interact with the Content Object Service or the Assembly File Service. Therefore, they can work independently but also make their content available in Visual Object repositories for others to use.

4.2.3 Delivery Architecture

Although this research does not offer specific contributions in the area of multimodal graphic interaction or presentation, the framework must be capable of supporting the delivery of multimodal graphics. Figure 4.2 shows the MGADF framework as it might be harnessed by a Delivery Interface. During delivery only a Composition Engine is required as new content is not being created. In addition, the Identifier Service is not contacted as all components already contain identifiers in order for them to be retrieved by the Delivery Interface. The ability to retrieve content from a service at runtime allows for dynamic delivery possibilities. Dynamic delivery refers to the realtime retrieval of content when requested by the user.

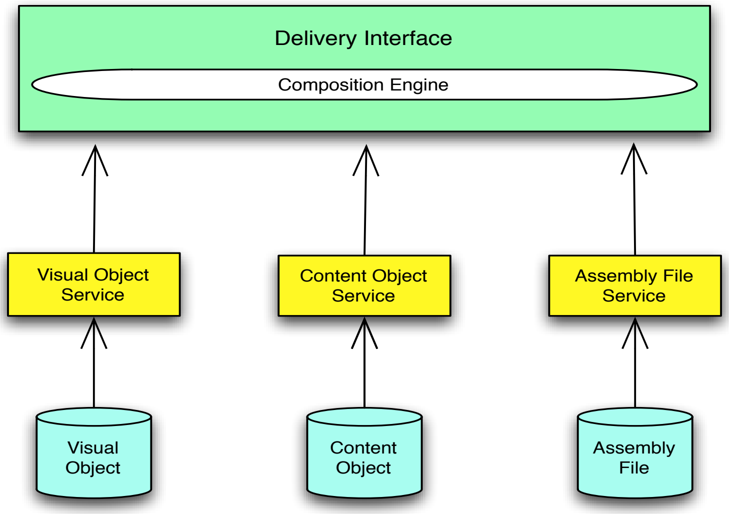


Figure 4.2: Delivery Architecture

For example, consider a learner interacting with a map of the USA, receiving supplementary geographical data about each state. Another tactile graphic may be available that deals with climate data in the USA that a learner would now like to interact with. As both tactiles are based on the same visual component, the learner can keep the same tactile graphic on their touchscreen, the content components relating to geography are unloaded and replaced with content components relating to climate. This functionality allows a learner to become familiar with a specific tactile graphic allowing them to focus on the supplementary information, a technique recommended in [Sheppard and Aldrich, 2001]. The dynamic nature of content delivery described above can be compared to the approaches advocated in systems such as KnowledgeTree discussed previously.

4.3 Information Architecture

The fourth objective of the research was to, "Design data models that can be used to represent the independent components of a multimodal graphic and provide interoperability". In order to conform to the system architecture discussed previously, models are required for each independent component of the framework. Each model must provide a mechanism for their identification, in order for them to be combined into a coherent multimodal graphic when requested by an interface. Three data models have been produced to satisfy the requirements; a Visual Object model, a Content Object model and an Assembly File model. The design of the information architecture was influenced by content reusability frameworks that exploit reusable learning objects. As there are no inherent learning goals in the components of a multimodal graphic, the terms Visual Object, Content Object and Assembly File are used. Each model must be based on an open data standard to facilitate interoperability. An XML Schema Definition [W3C, 2004b] has been designed for each model as outlined in detail in the following sections.

4.3.1 Metadata

Each model supports the addition of metadata to provide information relating to the object being modelled. This information can be used to identify an object for retrieval and to facilitate searching of the content in a repository. As there are no inherent learning goals in each object, data models from Technology Enhanced Learning, such as the IEEE LOM

[IEEE, 2002], contain elements that would not be useful for our approach. A simple form of data description was required, thus the Dublin Core Metadata [DCMI, 2004] approach was chosen. The data models in the framework support all 15 Dublin core elements; Contributor, Coverage, Creator, Date, Description, Format, Identifier, Language, Publisher, Relation, Rights, Source, Subject, Title, Type. The elements of Dublin Core that are required differ from object to object and will be elaborated upon in the following sections.

For the purpose of scope, this research assumes that the MGADF repositories will be filled with a small data set of Visual Object, Content Object and Assembly File models. Therefore, it is reasonable to assume that a user will be capable of easily identifying relevant content, by means of a keyword search of a given repository. Should the approach be implemented in a scenario where a larger data set is present, advanced information retrieval techniques, such as preselection through ranking and matching would need to be investigated. It should be noted that this would not affect the design of the MGADF but would enhance the efficiency of the approach.

4.3.2 Visual Object Model

The Visual Object model has been designed to represent all necessary information relating to a Visual Object. A Visual Object is a simple type. All that is required is metadata to describe the object and the location of the relevant image file so that it can be displayed on screen and printed onto a tactile. Only two XML child tags are required in the model, `<metadata>` and `<content>`. In order to accurately search for a Visual Object, the following metadata child elements are required in each Visual Object. In order to comply with the model, Visual Objects should be created with title, subject, description, creator, identifier and format Dublin Core Metadata elements. Title, subject, description and creator elements are useful for providing anyone running a search with an overview of what each object relates to. The identifier element is used in the Assembly File in order to inform an interfaces composition engine of which Visual Object should be retrieved from a repository for a given multimodal graphic. The format element is used to inform any interface that wishes to interact with a Visual Object of the type of image that it represents. The MIME (Multipurpose Internet Mail Extensions) Type [w3schools, 2010] of the image is used as content for this element. For example, if an SVG [W3C, 2010] image is being used, the format element will contain the value “image/svg+xml”. An interface examining this element would know that it must contain functionality to parse and render SVG in order to display the Visual Object. The use of a MIME type allows for the Visual Object model to be generalised to accept other image formats. No language element is used in the Visual Object model. During the period of formative research, audio tactile creators attempted to use the same image in multiple languages. The Visual Component creator had labelled elements of the image in a specific language. Due to this, the Visual Component was only useful in one language. Therefore, it is recommended that no textual labels appear in a Visual Component in order to enhance the reusability of the image. Only one `<metadata>` tag should exist in the model.

```
<vo xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xsi="http://www.w3.
    org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www
    .computing.dcu.ie/~dmcmullen/schema/vo.xsd">
    <metadata>
        <dc:title>The Human Digestive System</dc:title>
        <dc:subject>Biology</dc:subject>
        <dc:description>Tactile of the Digestive System</dc:description>
        <dc:creator>Declan</dc:creator>
        <dc:identifier>ie-dcu-vo1</dc:identifier>
        <dc:format>image/svg+xml</dc:format>
    </metadata>
    <content>http://www.computing.dcu.ie/~dmcmullen/svg/digestion.svg</
        content>
</vo>
```

Listing 4.1: Visual Object XML

The content tag must contain a URI that points to the location of the image file. The file can exist on a local machine, a file server or the Internet. When a Visual Object is to be loaded by an interface, it can retrieve the relevant file from the location pointed to by the `<content>` tag and render it on screen. Only one `<content>` tag should exist in the model. The XML Schema Definition for a Visual Object can be seen in appendix A.1. A sample XML implementation of a Visual Object data model is shown in listing 4.1

Prototype systems were provided to AHVITED project partners during the period of formative research. As the approach was required to mimic the T3 [TouchGraphics, 2010a] interaction strategy, audio tactile creators were required to manually annotate imported images. Project partners expressed concerns over the usability of the approach and the workload it added to the audio tactile creation process. Therefore, the MGADF requires the use of pre-annotated hit testable image formats. This means a format that allows an assembler to import an image and immediately click on elements in order to link Content Objects to them. No format should be used that requires a separate annotation procedure to annotate them. Additionally, each element in the image should have a unique identifier attached to it. This identifier will be used in the Assembly File in order to map elements of the image to specific Content Objects. This will be explained when describing the Assembly File Model in section 4.3.4.

It should be noted that the Visual Object model is not used to represent the annotated regions of a specific image. The annotations must be done by an external program and be present in the image file when it is provided to the system. The approach advocated in this thesis takes place when a suitable image has already been created and annotated. In future iterations of the approach, the need to model and store Visual Object annotations may be necessary. This is discussed in section 7.5

4.3.3 Content Object Model

The Content Object model has been designed to represent information relevant to all Content Objects. Three tags appear in the model; `<metadata>`, `<title>` and `<fragment>`. The following metadata elements are required in a Content Object; title, subject, description, creator, identifier and language. The title, subject, description, creator and identifier elements perform the same function as they do for Visual Objects. The language element is required in order to support localisation of Content Objects. A content component creator may wish to provide their content in multiple languages. If an assembler is trying to create a multimodal graphic in a given language, they should be able to search for suitable Content Objects in that language. The `<title>` tag can contain textual data representing the title of the Content Object. The contents of `<title>` will be played to the learner on their first press of the region that this Content Object is linked to. Only one `<title>` tag can be present in the model. The final tag in the model is the `<fragment>` tag. A Content Object model can contain numerous `<fragment>` tags. Each `<fragment>` tag contains an attribute called “mime-type” which defines the MIME type of a given fragment. As multiple fragments are permissible, it is possible for a mix of types to exist, for example, two fragments of text and a fragment of audio. The ability to mix MIME types is the reason that the format element was not used in the Content Object metadata.

```
<co xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xsi="http://www.w3.
    org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www
    .computing.dcu.ie/~dmcmullen/schema/co.xsd">
    <metadata>
        <dc:title>The Liver</dc:title>
        <dc:subject>Biology</dc:subject>
        <dc:description>The livers role in the human digestive system</
            dc:description>
        <dc:creator>Declan</dc:creator>
        <dc:language>en_GB</dc:language>
        <dc:identifier>ie-dcu-co1</dc:identifier>
    </metadata>
    <title>The Liver</title>
    <fragment mime-type="text/plain">
        <content>The liver produces bile that breaks up fats into
            droplets small enough to be digested</content>
    </fragment>
    <fragment mime-type="audio/mpeg">
        <content>http://atcdf.computing.dcu.ie/~dmcmullen/mgadf/files/
            liver2.mp3</content>
    </fragment>
</co>
```

Listing 4.2: Content Object XML

It was noted during the formative research that localising audio tactile content took longer when the creator chose to avail of recorded audio. Supplementary information had to be recorded for each language in which the audio tactile was to be delivered. In comparison, when layers of textual information were provided, localisation could be performed by typing in the equivalent information in a new language. Therefore, it is recommended that only textual fragments be used in order to support simple localisation. The Content Object Model can support multiple fragment types should the creator wish to provide them. The "mime-type" for a textual fragment is "text/plain". Each `<fragment>` tag contains a single `<content>` child tag. Only one `<content>` tag is permissible for each `<fragment>` tag. A value of the `<content>` tag is either some textual information or the URI to a type of file listed in the "mime-type" for the fragment. The XML Schema Definition for a Content Object can be seen in appendix A.2. A sample XML implementation of a Content Object data model is shown in listing 4.2

4.3.4 Assembly File Model

The Assembly File Model is the most detailed of the three models. It represents the metadata for a completed multimodal graphic, the location and identifier of each Visual and Content Object, the link between regions of the Visual Object and relevant Content Objects and the suggested sequence that a learner should follow when interacting with the multimodal graphic. The model contains `<metadata>`, `<components>`, `<mapping>` and `<sequence>` tags. The required metadata elements for an Assembly File are; title, subject, description, creator, language and identifier. These metadata elements are the same as those of a Content Object. The `<metadata>` tag can only exist once in an Assembly File.

The `<components>` tag contains information on the identifiers and locations of each component required by a composition engine in order to build a coherent multimodal graphic. The `<components>` tag contains a `<component>` child tag. There is no limit to the amount of `<component>` tags that may exist. Each `<component>` tag has two attributes "id" and "type‘". The "id" attribute contains the identifier of the object that the component relates to. The "type" attribute denotes the type of object to which the component relates. Available options are "vo" for Visual Object or "co" for Content Objects. An Assembly File should only have one Visual Object component but there is no limit to the number of Content Object components. Each `<component>` tag can have a single child tag called `<service>`. This tag is used to provide details about the service that is used to retrieve the parent component. A `<service>` tag has three attributes; "name", "namespace" and "wsdl", whose values should point to the name, namespace and WSDL (Web Services Description Language) location of the required service. This information can be used by an interface to retrieve the object of a specified type with a specified identifier using a specified Web Service.

The `<mapping>` tag is used to provide information on the link between the elements of a Visual Object and the Content Objects that relate to those elements. A single child tag called `<map>` is used. The `<map>` tag contains two attributes "coId" and "voElementId". The value of the "coId" attribute contains an identifier of a Content Object component that appears in the `<components>` tag above. The value of the "voElementId" contains the identifier of a specific element in the Visual Object component provided in the `<components>` tag above. For example, consider that a Visual Object of the Human Digestive System contains a visual element, with an identifier of "element1", that represents the Liver. A Content Object for the Liver is available with the identifier "ie-dcu-co1". The map information to link the two together would be `<map coId="ie-dcu-co1" voElementId="element1"/>`. A `<map>` tag should exist to link each Content Object component that is defined in the `<components>` section with an element of the visual component. This mapping of elements to content objects is illustrated in figure 4.3

Previous audio tactile systems, Nomad [Parkes, 1988], T3 [TouchGraphics, 2010a] and IVEO [ViewPlus, 2010a] have all provided learners with the ability to receive guided instruction when exploring a tactile graphic. A learner is provided with a list of all the elements that exist in the graphic. They can then select a specific element that they wish to have their finger guided to. As the learner presses on the graphic, instructions, such as "go up" or "go down", are played in order for them to be guided to a specific element. The same functionality can be used to guide the learners interaction with the graphic in order to ensure that elements are touched in a specific order. Support for that functionality was to be maintained in the data models of this research. Therefore the ability to represent a sequence of graphical elements was required.

The `<sequence>` tag is used to provide sequencing information to an interface, which can aid a learner in interacting with the multimodal graphic in a suggested order. A child tag called `<coref>`, which contains a single attribute called "id", is used to define the sequence. The value of the "id" attribute contains the identifier of a Content Object defined in the `<components>` section. A delivery system can use the sequence of the `<coref>` tags to control the order in which the learner explores the graphic, as discussed in the previous paragraph. There is no limit on the number of `<coref>` tags that can be used but generally each Content Object should be included.

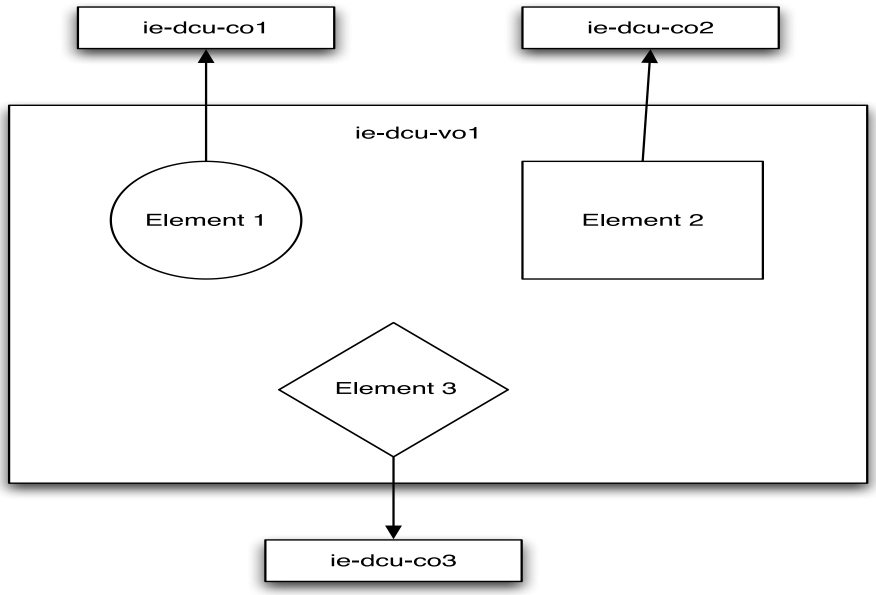


Figure 4.3: Visual Object Element to Content Object Mapping

No specific MIME type information is provided in the Assembly File. As there are so many combinations of Visual Object types and Content Object fragment types no one MIME type would suffice. This compares to the IMS Content Packaging (IMS CP) standard which allows adopters to "gather, structure and aggregate content in an unlimited variety of formats" [IMS, 2007]. Therefore, it is recommended that interface implementers use the Assembly Files in order to query the MIME types of the Visual Object and Content Objects that are required to compose a coherent graphic. Once performed, the interface will know whether it supports each component and how to render it. If any part of the multimodal graphic is not supported by the interface, the learner can be informed.

The XML Schema Definition for an Assembly File can be seen in appendix A.3. A sample XML implementation of an Assembly File Model is shown in listing 4.3.

```
<assembly xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xsi="http://
    www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="
    http://www.computing.dcu.ie/~dmcmullen/schema/assembly.xsd">
    <metadata>
        <dc:title>Human Digestive System</dc:title>
        <dc:subject>Biology</dc:subject>
        <dc:description>multimodal graphic of Human Digestive System</
            dc:description>
        <dc:creator>Declan</dc:creator>
        <dc:language>en_GB</dc:language>
        <dc:identifier>ie-dcu-af1</dc:identifier>
    </metadata>
    <components>
        <component id="ie-dcu-vo1" type="vo">
          <service name="VisualObjectImplService" namespace="http://
              service.visualobjectservice.atcdf/" wsdl="http://atcdf.
              computing.dcu.ie:8080/VisualObjectService/
              VisualObjectImplService?wsdl"/>
        </component>
        <component id="ie-dcu-co1" type="co">
          <service name="ContentObjectImplService" namespace="http://
              service.contentobjectservice.atcdf/" wsdl="http://atcdf.
              computing.dcu.ie:8080/ContentObjectService/
              ContentObjectImplService?wsdl"/>
        </component>
    </components>
    <mapping>
        <map coId="ie-dcu-co1" voElementId="element1"/>
    </mapping>
    <sequence>
        <coref id="ie-dcu-co1"/>
    </sequence>
</assembly>
```

Listing 4.3: Assembly File XML

It should be noted that the integrity of the information models is not considered at this time. For example, imagine components that are listed in a given Assembly File are deleted from their respective repositories. The Assembly File is not aware of these removals and therefore dangling references are left to content which no longer exists. Future work should consider a form of validation which can assess the information models for integrity and provide feedback to the user.

4.4 Methodologies

It should be noted that in this thesis the term “methodology” refers to the process employed by the end user in order to complete a specific multimodal graphic assembly task. Dagger states that “typically a course construction methodology consists of six high level phases, namely, analysis, planning, designing, developing, implementing and evaluating” [Dagger, 2006]. The methodology of the MGADF takes place in the content development phase, which consists of the identification and selection of appropriate content. The “reuse of existing material and services could help to reduce workload and increase efficiency of this development phase” [Dagger, 2006]. Therefore, the assembly methodology of the MGADF consists of locating and combining existing content into coherent multimodal graphics.

The fifth objective of the research was to, “Design a methodology for combining independent components into a coherent multimodal graphic”. The following section describes various methodologies for how the framework can be used. Visual Component Creators, Content Component Creators and Multimodal Graphic Assemblers are all taken into account in the design of the methodologies. Visual Component Creators are those who make suitable graphical images available for a multimodal graphic to be based on. Content Component Creators provide the supplementary information that can be attached to elements of the multimodal graphic. Multimodal Graphic Assemblers are those who combine Visual and Content Components to facilitate the assembly of a multimodal graphic.

4.4.1 Visual Component Creator

Visual Component Creators can use the drawing package of their choice in order to create images suitable for use as tactile graphics. The images should be saved in a pre-annotated format as described in section 4.3.2. Creators should be provided with an interface whose sole functionality is to provide access to a Visual Object Service and in turn a Visual Object Repository. The interface should provide the creator with the ability to enter details about one or more Visual Object Services. In order to store a Visual Object, the creator should import one of their image files into the interface. They should then provide metadata for the image as required by the Visual Object Model described in section 4.3.2. Once all of the information has been provided, the creator should inform the interface to save the Visual Object using a specific service. An Identifier Service should be queried in order to retrieve a unique identifier for the object being saved. The information can now be passed to the Visual Object Service and stored in a Visual Object Repository. The creator should be allowed to search for Visual Objects that already exist in the repository and retrieve them for editing.

4.4.2 Content Component Creator

It is recommended that Content Component Creators provide textual content fragments for use in their Content Objects. If non textual fragments are being used, they should be created using an appropriate program. Creators should be provided with an interface whose sole functionality is to provide access to a Content Object Service and in turn a Content Object Repository. The interface should provide the creator with the ability to enter details about one or more Content Object Services. In order to create a Content Object, the creator should enter a title and a number of content fragments. They should then provide metadata for the content as required by the Content Object Model described in section 4.3.3. Once all of the information has been provided, the creator should inform the interface to save the Content Object using a specific service. An Identifier Service should be queried in order to retrieve a unique identifier for the object being saved. The information can now be passed to the Content Object Service and stored in a Content Object Repository. The creator should be allowed to search for Content Objects that already exist in the repository and retrieve them for editing.

4.4.3 Multimodal Graphic Assembler

There are three distinct variations on the methodology that can be employed by a multimodal graphic assembler. An assembler may have all of their own content already available and therefore does not require any support. This is more likely to be used by an experienced multimodal graphic assembler. Alternatively, and more likely for non-experts, an assembler would like to search for pre-existing content upon which to base their multimodal graphic. Lastly, a combination of the two methods is possible where an assembler can reuse a Visual Object but add their own Content Objects or vice versa. It should be noted that it is possible to use an entirely pre-assembled multimodal graphic by harnessing an existing Assembly File but as no assembly would be taking place the scenario is not discussed in the following sections. An activity diagram representing the assembly process can be seen in figure 4.4.

In each scenario an Assembly Interface should be provided with the following functionality. The assembler should be able to enter the details about a single or numerous Visual Object, Content Object and Assembly File services. The assembler should be able to create, search for or reuse existing Visual and Content Objects. The assembler should also be able to enter metadata about the multimodal graphic they are assembling. The interface should contain functionality to store the components of the assembled graphic using the selected services.

4.4.3.1 No Component Reuse

In this scenario an assembler does not wish to use pre-existing content. The assembler begins by creating a Visual Object. The assembler takes on the role of a Visual Component Creator. They should import a suitable image file and enter appropriate metadata. The assembler should be able to choose the service that will be used to save the Visual Object. Once complete, the Visual Object can be temporarily stored in order for content components to be linked to it. The assembler should click on the element of the graphic that they wish to link a Content Object with. The assembler is now assuming the role of a Content Object Creator. They should enter a title and a number of fragments for the Content Object they are creating. They should then enter metadata to describe the object they are creating. The assembler should be able to choose the service that will be used to save the Content Object. This process can be repeated for further elements of the graphic that an assembler would like to link Content Objects to. Once complete, the assembler should add metadata for the entire multimodal graphic. The interface should allow for all required metadata elements as outlined in section 4.3.4 to be entered. The assembler should be provided with functionality to set a sequence for the Content Objects that the learner should follow whilst exploring the graphic. An assembler should be able to save their multimodal graphic by selecting the Assembly File Service that will store the Assembly File.

4.4.3.2 Complete Component Reuse

The following scenario is the primary scenario at which this research is aimed, non-expert multimodal graphic assemblers who need to be supported in the simple creation of multimodal graphics. It is assumed that the assembler does not have any content of their own that they wish to use. Each multimodal graphic must be based on a Visual Object and as such the first step in this methodology is to locate a suitable Visual Object. The assembler should be provided with functionality to choose a Visual Object Service and search the contents of the repository it is configured to access. Once an assembler has located a suitable Visual Object to form the base of their multimodal graphic, it should be retrieved from the repository and imported into the Assembly Interface. The assembler should now attach Content Objects to the desired elements of the Visual Object for which they wish supplementary information to be available. Functionality should be available to search for Content Objects using a selected Content Object Service. Once a suitable Content Object is found that the assembler wishes to link to a given region, it can be imported into the Assembly Interface. The assembler should continue the process of searching for suitable Content Objects until they have linked objects to each region of the Visual Object for which they wish supplementary information to be available. In order to complete the multimodal graphic, the assembler should add values for all required metadata elements as outlined in section 4.3.4. This is the first point in the methodology where pre-existing information is not available. The assembler should be provided with functionality to set a sequence for the

Content Object that the learner should follow whilst exploring the graphic. An assembler should be able to save their multimodal graphic by selecting the Assembly File Service that will store the Assembly File.

4.4.3.3 Mixed Component Reuse

The final scenario involves a mixture of the previous two scenarios. An assembler may have some content they wish to provide and other content they wish to locate via a search. In this scenario we will assume that the assembler wishes to provide their own Content Objects but reuse an existing Visual Object. The assembler should search for a suitable Visual Object using a selected Visual Object Service. Once one has been located it should be imported into the Assembly Interface. The assembler can now create new Content Objects that can be linked to specific regions of the Visual Object. Once supplementary content has been added to all required regions, the assembler should enter the metadata for the entire multimodal graphic. The assembler should be provided with functionality to set a sequence for the Content Objects that the learner should follow whilst exploring the graphic. An assembler should be able to save their multimodal graphic by selecting the Assembly File Service that will store the Assembly File.

4.4.3.4 Storage

Once an assembler has chosen to save their multimodal graphic, the decomposition engine should take control and store each multimodal graphic component independently using the services selected by the assembler. Any new objects should be created and stored in the relevant repository. Any reused objects are left in the repository they were retrieved from and no replication of that object takes place. Unless a multimodal graphic is being edited, a new Assembly File should be created when a multimodal graphic is saved. The Assembly File should contain the metadata entered by the assembler and the identifiers and locations of each object as described in section 4.3.4. Once the decomposition process is complete, the components are not only reusable by the assembler but also by other multimodal graphic assemblers with access to the same services.

4.4.3.5 Editing

Assemblers should be provided with functionality to edit existing multimodal graphics. In order to do so, the Assembly Interface should allow an assembler to search, using a selected Assembly File Service, for completed Assembly Files. If the assembler wishes to edit the multimodal graphic, the Assembly File should be selected. The composition engine should populate the multimodal graphic metadata, retrieve the Visual Object using its relevant service and retrieve each Content Object using its relevant service. A coherent multimodal graphic is now available for editing by the assembler.

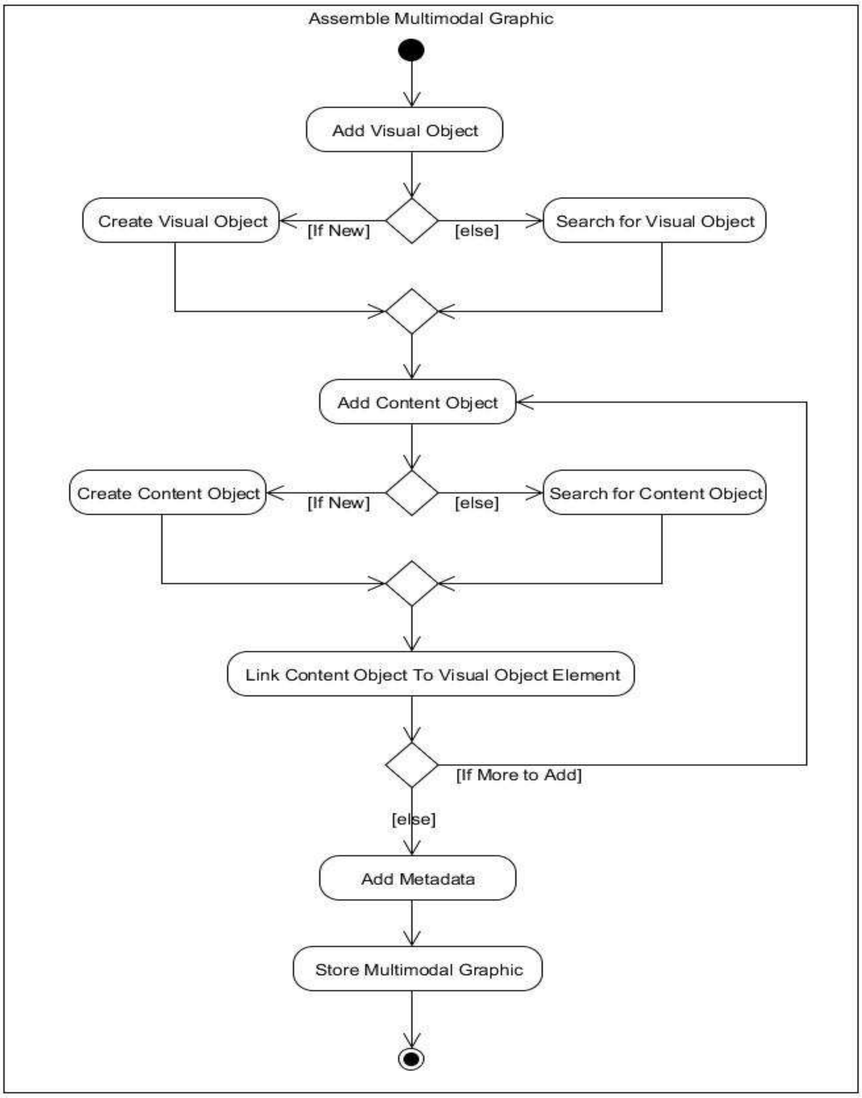


Figure 4.4: Multimodal Graphic Assembly Activity Diagram

4.5 Summary

This chapter discussed the design of the Multimodal Graphic Assembly and Delivery Framework (MGADF). Section 4.2 discussed how specific techniques from Technology Enhanced Learning influenced the design of the MGADF system architecture. Section 4.3 outlined the design of the MGADF Information Architecture and detailed the data models that represent each system component. Finally, section 4.4 detailed numerous methodologies that can be followed to exploit the capabilities of the MGADF. The interaction of Visual Object Creators, Content Object Creators and Multimodal Graphic Assemblers were discussed.

Chapter 5

Implementation

5.1 Introduction

The sixth objective of the research was to, "Implement a proof of concept system that clearly demonstrates the feasibility of the approach". This chapter provides an overview of the proof of concept systems that serve as an implementation of the frameworks architecture, data models and methodologies, in order to support a summative evaluation. This phase of the work relates to the "Visualization Component" of the generic architecture [Siqueira et al., 2003], in particular it's "Interface" element, discussed in chapter 2. Numerous implementations took place in order to fully support the framework. Each service, Visual Object, Content Object, Assembly File and Identifier was implemented and suitable object repositories created. Interfaces were created for Visual Component Creators, Content Component Creators and Multimodal Graphic Assemblers. Methodologies for each user type were discussed in section 4.4

In section 5.2 the technologies used in the Web Services, repositories and user interfaces are described. Section 5.3 discusses the implementation of the services and repositories relating to the System Architecture described in chapter 4. The user interfaces of the MGADF are discussed in section 5.4. The interfaces are capable of supporting the methodologies described in chapter 4. Examples and screenshots are provided to illustrate interface functionality.

It must be stressed that this is a single proof of concept implementation of the framework. Other researchers could implement the framework in their own way and the abstract nature of the framework allows them to do so. The term multimodal graphic is used as the framework should be able to support other forms of accessible graphic display and interaction. It should be noted that the interfaces developed during this research are suitable for touchscreen based audio tactile graphics.

5.2 Technology Overview

Siqueira stated that "in a real world deployment of an architecture for education and training systems based on web technology, it is important to provide loosely coupled, component-oriented and cross technology implementations" [Siqueira et al., 2008]. The LORIS architecture [de Moura et al., 2005], that serves as an implementation of the generic architecture [Siqueira et al., 2003] uses cross platform technologies such as JAVA [JAVA, 2010], SOAP(Simple Object Access Protocol) [W3C, 2007a], WSDL(Web Services Description Language) [W3C, 2007b] and XML(Extensible Markup Language)[W3C, 2008]. The same technologies are harnessed in the MGADF proof of concept implementations.

Figure 5.1 shows the architecture of the framework with the relevant technologies listed. Each desktop interface was implemented using the JAVA programming language and the SWING graphical user interface toolkit [Hoy et al., 2002]. Text to speech is supported using the JAVA Speech API (JSAPI) [JSAPI, 2010] to communicate with the Windows Speech API (SAPI) [SAPI, 2010]. Each Web Service was also implemented in JAVA using the JAX-WS Application Programming Interface [JAX-WS, 2006]. The repositories for the Visual Object, Content Object and Assembly File services are facilitated using the eXist database management system [eXist, 2010]. eXist is specifically designed to cater for XML data and is suitable for the persistence of the MGADF data models. The identifier repository is facilitated using the MySQL relational database management system [MySQL, 2010]. The services communicate with their repositories using the relevant JAVA API's necessary to interact with the chosen database backends. The services communicate with the graphical user interfaces using HTTP [HTTP, 2010] and SOAP [W3C, 2007a]. The Visual Object, Content Object and Assembly File services also communicate with the Identifier service using HTTP and SOAP. The methods available for use in each service are exposed to the user interfaces using the Web Service Description Language (WSDL) [W3C, 2007b]. All services in the proof of concept system are deployed to and accessed from a Glassfish [GlassFish, 2010] application server. The E-Learning integration implementation is in the form of a plug-in for the open source Virtual Learning Environment Moodle [Moodle, 2010]. The Moodle instance is hosted on an Apache web server [Apache, 2010a]. The plug-in was developed using the PHP scripting language [PHP, 2010]. The Glassfish and Apache servers are hosted on the same server running Ubuntu Linux [Ubuntu, 2010].

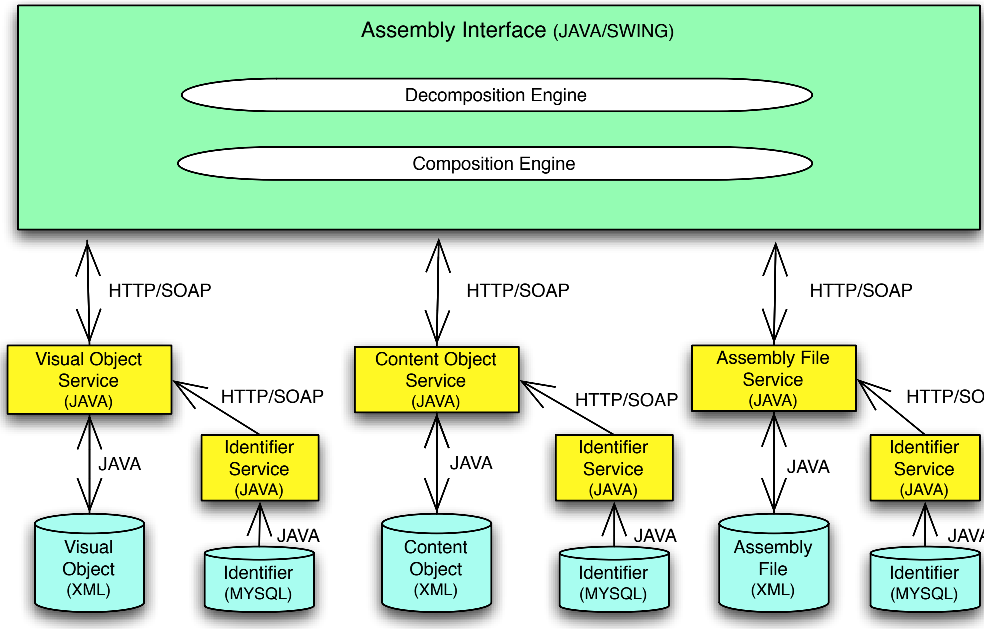


Figure 5.1: Technical Architecture

It is possible to implement the framework using various levels of granularity. For example, each Web Service could be hosted on different servers in different locations. The instance of the backend repository for a service need not be located on the same server as the Web Service itself. A single Identifier Service implementation can be used to provide identifiers to each service or each service can interact with their own Identifier Service implementation. For the purposes of proof of concept, every Web Service and every repository resides on the same server for this implementation of the framework. In addition, a single Identifier Service is shared amongst the Visual, Content and Assembly File services.

Although JAVA and PHP are used in our proof of concept implementations, the use of open standards such as XML, HTTP and SOAP, allow any programming language capable of supporting those protocols to be used to develop interfaces that interact with the MGADF. A mix of multiple operating systems could be used to host and interact with the components of the framework. Specific details of the implementation will be outlined in the following sections.

5.3 Services and Repositories

5.3.1 Identifier Service

The Identifier Service exposes a single method to any interface or service that wishes to interact with it. The method, called getUniqueIdentifier, takes three parameters, constructs an identifier based on those parameters and returns the identifier to the interface or service that requested it. Instead of designing our own identification scheme, an existing design from the area of accessibility is used. Identifiers in the MGADF follow the same scheme as the DAISY 3 specification [DAISY, 2010]. An "identifier under this scheme consists of a hyphen-separated string consisting of a two-letter country code drawn from [ISO 3166], an agency code unique within it's country, and an identifier unique within the agency" [DAISY, 2005].

The valid parameters for the getUniqueIdentifier method are; a string that denotes a country code, a string that denotes an institution code and an integer that denotes the type of identifier to be constructed. An integer of "1" relates to a Visual Object, "2" a Content Object and "3" an Assembly File. The layout of the identifier is a hyphen separated string of country code, institution code and unique identifier. The unique identifier contains a combination of a code to signify the type of identifier and a unique digit. The unique codes are "vo" for Visual Object, "co" for Content Object and "af" for Assembly File. A sample identifier that would be constructed and returned by the identifier services is "ie-dcu-co1". The identifier represents a content object created in DCU Ireland.

The identifier repository was implemented using MySQL. A table was created for each type of identifier containing a single integer. The integer is used to keep track of the unique digits that have been previously used. This ensures that an identifier can never be duplicated.

After an identifier of a given type is created, the integer in the table related to that type of identifier is incremented.

5.3.2 Component Services

5.3.2.1 Data Model Management

In order to reduce the complexity of the proof of concept interfaces, a development decision was taken that the services and user interfaces should not pass raw XML data models back and forth. Instead, data structures (Objects) would be developed to contain all of the attributes and methods required for a given multimodal graphic component, namely Visual Objects, Content Objects and Assembly Files. Communication between interfaces and services therefore takes place by passing objects of those types back and forth. These objects can be created using any programming language as long as they provide the attributes required by the services that receive them.

In addition to communicating via objects, the construction of relevant XML data models is implemented in each service. For example, when a service receives an object that is to be stored, it uses the data in the object to create an XML data model representation of the object and stores it in the relevant repository. In order to return the data to an interface that requests it, the XML data model is parsed, an object is created to match the content and it is sent to the interface that requested it. This approach makes it easy for various graphical user interfaces to be developed for interacting with the framework as they do not need to build or parse XML that conforms to the MGADF data models. In addition, as the services are building the data models, we did not need to implement a data model validation procedure. The interfaces cannot pass XML data models to the services and as such cannot provide non validating XML.

The eXist database management system is used as a backend for each object repository. Data is grouped into Collections in a similar fashion to the use of tables in an SQL database. In our implementation, each object type has its own Collection in the eXist database. eXist provides Lucene index [Apache, 2010b] functionality, which is a high performance text search engine library. Using this library it is possible to run searches of the XML data models using an XQuery [W3C, 2007c] syntax for given keywords or identifiers. This is harnessed to expose methods to user interfaces that can be used to run keyword and known identifier searches of an objects metadata. The benefits of the functionality will be seen in the discussion of the user interfaces in section 5.4.

5.3.2.2 Exposed Methods

All three services, Visual Object, Content Object and Assembly File, will be discussed in this section. Each service shares the same methods and functionality and only differ in their data types and the use of some additional methods. Therefore, a generic explanation of the methods exposed by the services is provided in the following sections. Methods are provided to store an object, retrieve an object, delete an object, list all objects and search for a specific object.

The storeObject method can take two parameters, an object of the type compatible with the service and a boolean informing the service whether the object is new or not. The reason for the boolean is to allow for interfaces to edit previously created objects. If the object is new, the service should build an XML data model to match the information in the object and store it in the relevant repository. If an object is being updated, the new information for the object should be stored in the repository but the object should not be duplicated.

The getObject method takes a single parameter, a string that represents the identifier of the object the interface is requesting. Upon receiving the request, the service queries the repository for the data model containing the identifier provided. Once located the data model is parsed and an object of the relevant type is created containing the information in the data model. The object is then returned to the requesting interface.

In order to allow a user to remove unwanted or inaccurate content, an interface can also trigger the deletion of an object in the repository using a similar method. The deleteObject method takes an identifier and uses it to locate and delete the data model containing the identifier provided. The methods listAllObjects, searchForObjectByKeyword and search-ForObjectByIdentifier are provided to allow an interface to display all of the content in a given repository or search for specific entries. These methods are required in order to support a non-expert assembler in locating relevant content for them to reuse.

In order to reduce the complexity of the user interfaces, the retrieval of suitable identifiers for each object is performed by the relevant service. When a service is building the XML data model for an object, it queries an Identifier Service in order to receive a suitable identifier. The identifier is added into the metadata for the object and used for its filename in the objects eXist Collection. The alternative was to provide each user interface with functionality to retrieve an identifier prior to an object being sent to the relevant service for storage. In order to reduce the number or service calls being made by the interface and to reduce their complexity, the functionality resides in each service instead.

The proof of concept implementation of the Visual Object service contains an additional storage method called storeObjectContent. This method is designed to be used as a wrapper to the storeObject method and contains three parameters, a Visual Object, a boolean and a string. The first two parameters are the same as described above and are passed on internally to the original method. The third parameter contains the content of the image that the Visual Object is based on. Our proof of concepts deal primarily with SVG and as such the contents of the string contain SVG data for the image. The data is used to store a copy of the imported image on the server.

There are two reasons for doing so. The first is for cases where the image was not provided as a URI. The image file may be located on the assemblers computer. The file would not be reusable by other assemblers if it could only be retrieved from the assemblers computer. Therefore, some form of file server should be considered as part of the methodology if the file is not already publically available as a URI. Secondly, we discussed previously that the use of pre-annotated image formats is recommended in the framework. The elements in the image should contain identifiers so that they can be used to map the elements to Content Objects for delivery to the learner. As the use of identifiers on SVG elements is not compulsory, some SVG images do not contain identifiers. For this reason, our proof of concept system checks for element identifiers and adds them where they do not already exist. In order to persist the updated SVG file, it must be stored on the server. The method was added to the Visual Object service to provide makeshift fileserver functionality for convenience. It would not normally be required to form the implementation of a Visual Object service.

Each service has user configurable property files which allow the administrator to set the location of the Identifier Service and eXist server to use. The eXist property file can be seen in listing 5.1 and contains values for the XML database URL and Collection to be used for a given service. The Identifier Service property file can be seen in listing 5.2 and contains values for the WSDL URI, namespace and name required to connect to the service. Identifiers are generated in a specific format for each type of object. Therefore, the ability to set a custom locale code, institution code and identifier type that will be used by the identifier service is facilitated. The operation of the Identifier Service was discussed in section 5.3.1.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <comment>eXist Properties</comment>
  <entry key="xmldb.url">xmldb:exist://localhost:9090/exist/xmlrpc</entry
      >
  <entry key="xmldb.collection">/db/atcdf/visualobject</entry>
</properties>
```

Listing 5.1: eXist Property File

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <comment>Identifier Service Properties</comment>
  <entry key="service.wsdl">http://localhost:8080/IdentifierService/
      IdentifierImplService?wsdl</entry>
  <entry key="service.namespace">http://service.identifierservice.atcdf/<
      /entry>
  <entry key="service.name">IdentifierImplService</entry>
  <entry key="identifier.localeCode">ie</entry>
  <entry key="identifier.institutionCode">dcu</entry>
  <entry key="identifier.type">1</entry>
</properties>
```

Listing 5.2: Identifier Service Property File

5.4 User Interfaces

The methodologies described in section 4.4 are illustrated in the following sections for each user type. Screenshots of the proof of concept systems are used for illustration.

5.4.1 Visual Object Tool

The Visual Object Tool was designed to provide Visual Component Creators with the ability to store their components in a repository, thereby making them available for reuse. The Visual Object Tool can be set up to operate with multiple different Visual Object services that provide access to the relevant repositories. In order to specify the details of the services, the service panel can be used.

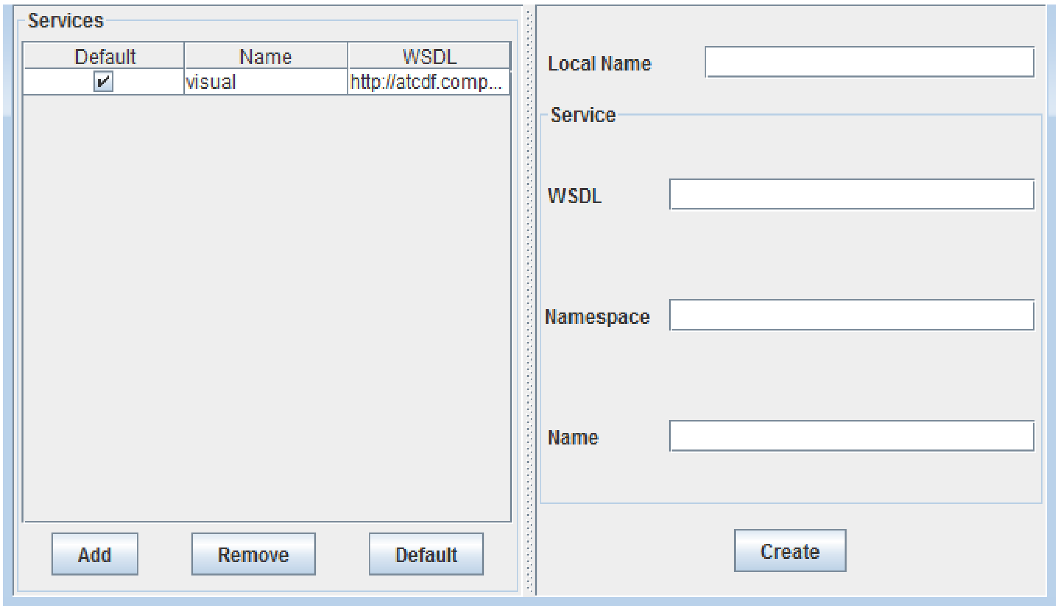


Figure 5.2: Service Panel

The panel can be seen in figure 5.2. The left side of the panel provides the user with a list of services that have already been specified. The local name given to the service for use in the interface and it's WSDL URI are shown in order to provide an overview of each service. A checkbox is available in order to choose a default service. The default service is used when a Visual Object is being saved, unless the user chooses a different service. The left side of the panel also provides buttons to add a service, remove a service and set a service as the default. If a user chooses to add or edit a service, the right side of the panel is used. New services can provide the assembler with access to additional repositories of objects. The right hand side of the panel contains a number of text boxes in order to enter information about a given service. The local name, name, namespace and WSDL URI of the service can all be entered. A create button is located at the bottom of the panel to allow the information to be persisted. If an existing service is being edited the button will say "Save".

Once at least one service has been provided for the Visual Object Tool to interact with, the user can either add a new Visual Object, search for a Visual Object or edit a Visual Object. In order to store a new Visual Object, the Visual Object panel is used, as can be seen in figure 5.3.

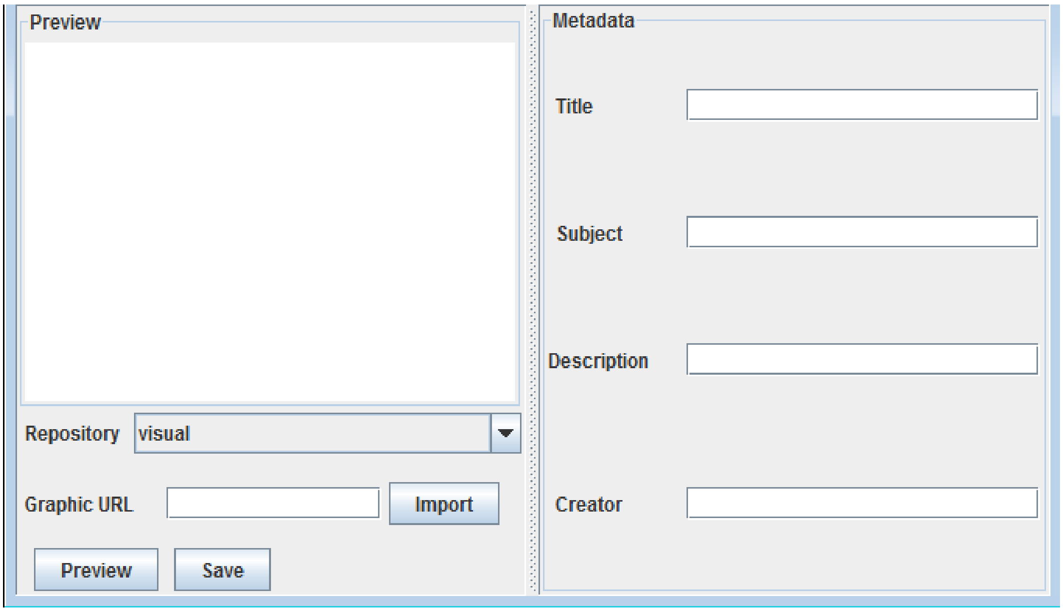


Figure 5.3: Visual Object Panel

The panel is split in two halves down the middle. The left side of the panel allows a user to select an image on which the Visual Object is to be based. The image can be previewed in the window at the top left of the panel. The left side of the panel also contains a dropdown menu containing the local names of each service that is available to the user. The default service will be used if another service is not chosen. The right hand side of the panel allows the user to enter metadata about the Visual Object being created. Text boxes for title, subject, description and creator are provided in order to comply with the Visual Object Model as defined in section 4.3.2. When the "Save" button is selected, the Visual

Object will be sent to the chosen service for storage in the relevant repository. The object is then immediately available for reuse by others with access to the chosen service.

In order to edit a Visual Object, a two phase procedure is provided. First, the user interacts with the search panel as seen in figure 5.4. The panel allows a user to see every Visual Object provided by a given service or search for a specific Visual Object. The search panel is split into three separate parts. The top part of the panel allows the user to alter the terms of the search. A single service can be chosen from the drop down menu. Radio buttons are provided to inform the service whether a search is be performed using keywords or using an object identifier. A textbox is provided in order to enter either the keywords or the identifier for the search. Alternatively, a button is provided in order to list every Visual Object available in the repository that can be accessed using the selected service.

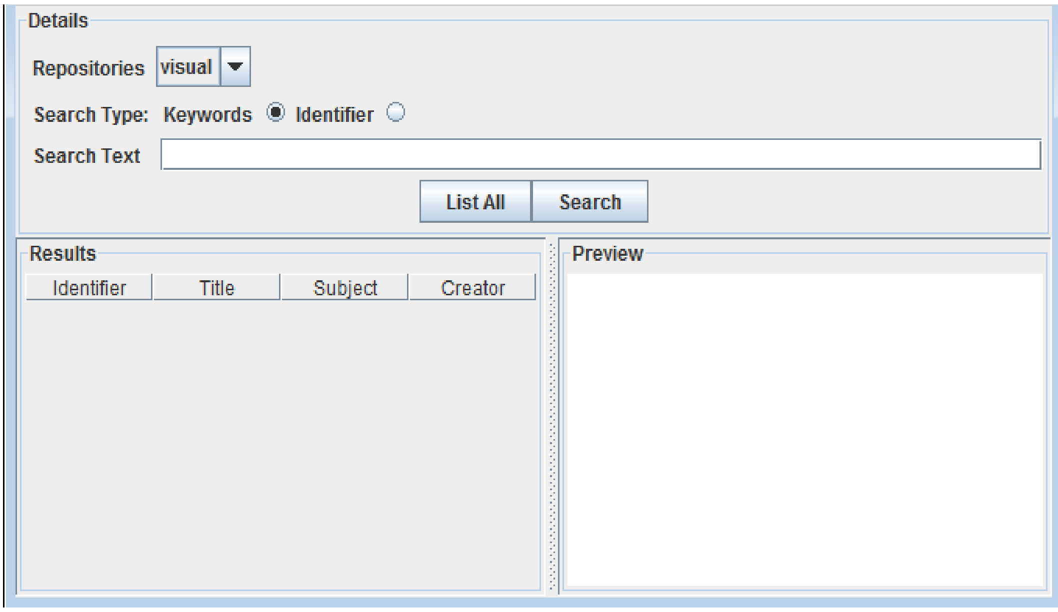


Figure 5.4: Search Panel

Once a search has been performed the results will appear in the bottom left hand side of the panel. The results are split into four columns with the headings, Identifier, Title, Subject and Creator. These pieces of metadata provide an overview of each Visual Object returned in the search. If a user would like to preview the image relating to a given Visual Object, they may click on the relevant entry in the result list and the required image will appear in the preview window on the bottom right of the panel. Once an object has been selected a user can either delete it or edit it. If they choose to edit it, the Visual Object panel will appear again but pre-populated with the information relating to the chosen object. The user can alter the elements of the object, such as the image or the metadata and resave the changes using the service. It should be noted that if an object is retrieved from a given service it will be resaved to that service. The user cannot alter the location to which an object can be stored if it already exists.

Using the Visual Object Tool, Visual Component Creators can operate independently but also make their content available for others to use.

5.4.2 Content Object Tool

The Content Object Tool operates in a similar fashion to the Visual Object tool and was designed to allow Content Component Creators to store Content Objects in repositories for others to reuse. Users follow the same process for entering Content Object Services as they do in the Visual Object Tool, by using the service panel seen in figure 5.2. Once services are available the user can create a new Content Object using the Content Object panel. The panel can be seen in figure 5.5.

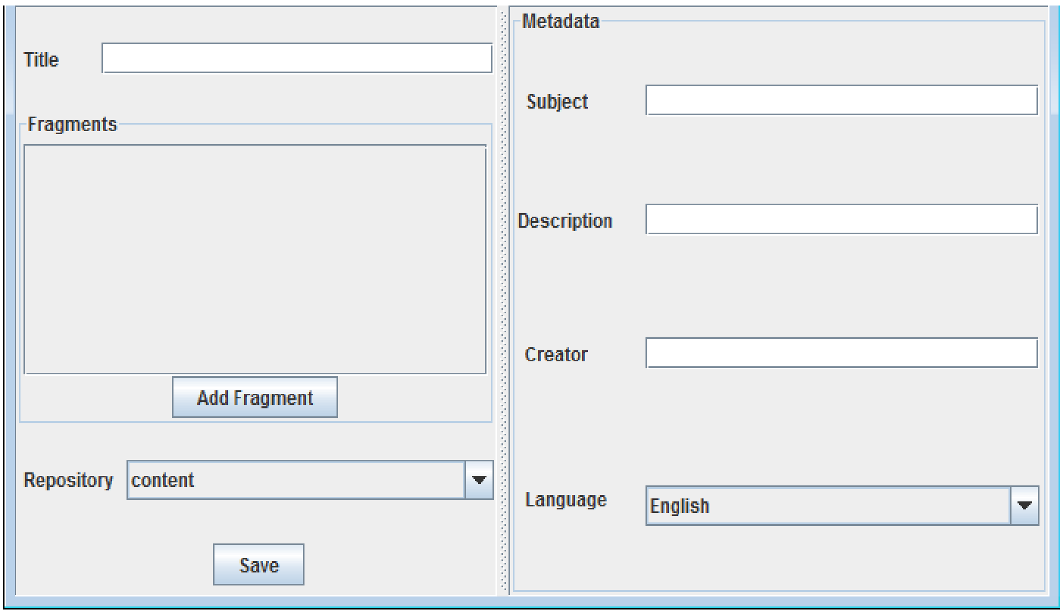


Figure 5.5: Content Object Panel

The left side of the panel allows a creator to enter the title and fragments for the Content Object they wish to create. Only textual fragments can be entered using the panel in order to enable simple localisation, as discussed in chapter 4. The dropdown to choose a service is located on the left side of the panel and, as before, the default service will be used if another service is not chosen. The right hand side of the panel allows the user to enter metadata for the Content Object. Text boxes are provided for subject, description and creator and a dropdown box is provided for language, in order to conform with the required metadata from the Content Object Model. Once the user selects “Save”, the Content Object will be permanently stored using the chosen service. The object is then immediately available for reuse by others with access to the same service.

The process of editing a Content Object is identical to that of editing a Visual Object. The only variation is that the search panels preview window will display the title and textual fragments of a Content Object rather than the image of a Visual Object. Using the Content Object Tool, Content Component Creators can operate independently but also make their content available for others to use.

5.4.3 Assembly Tool

This section will discuss the implementation of the Assembly Tool designed to support multimodal graphic assemblers. During the design discussions in section 4.4.3, three distinct variations of the assembly methodology where provided. The first method consisted of an assembler who had their own content available and did not require assistance in locating suitable content. This method is more suited to an experienced multimodal graphic assembler. The second method was that used by non-expert assemblers and involved the assembler being supported in locating suitable Visual and Content Objects for use in their multimodal graphic. The final method was a combination of both approaches, were the assembler wished to search for some content but also provide their own. Only the first two methods will be discussed in detail as the third method is a combination of both and can therefore be extrapolated from the information in the following sections.

5.4.3.1 Services

The assembler can avail of functionality to provide information on Assembly File Services where multimodal graphics can be stored. As multimodal graphics also consist of a Visual Object and multiple Content Objects, functionality is also available to provide information on Visual and Content Object Services. The service panel that can be used in the Assembly Tool can be seen in figure 5.6. This is a tabbed panel with a tab for each type of service running along the top of the panel. Once a given tab is selected the service panel it reveals is the same as previously described in sections 5.4.1 and 5.4.2. Multiple services can be added for each type and the default service will be used in the interface if no other service is chosen by the user. The following information assumes that services for each type of object have already been provided.

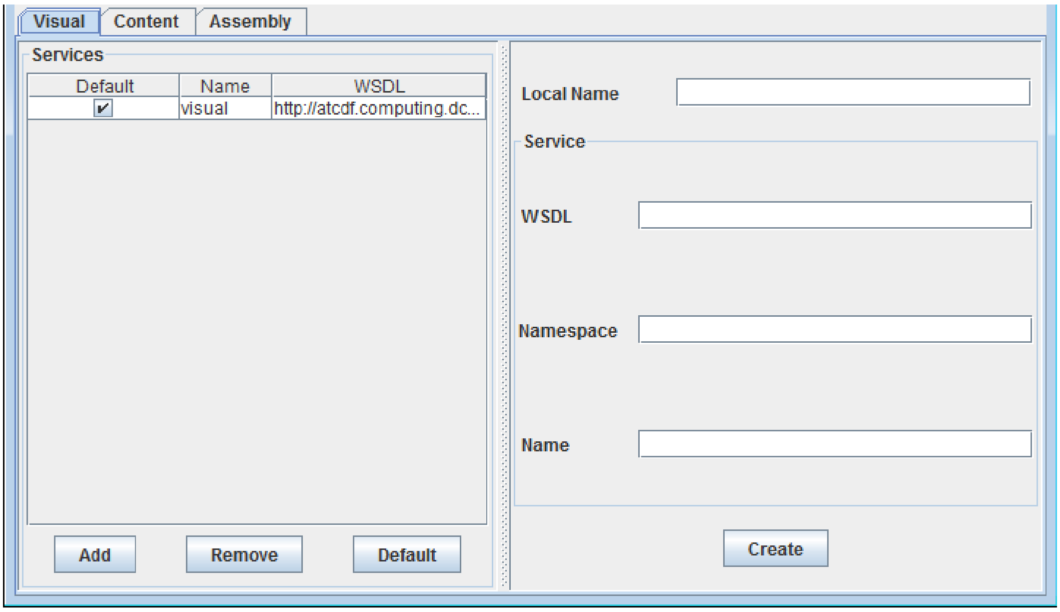


Figure 5.6: Assembly Tool Services Panel

5.4.3.2 No Component Reuse

As an assembler may be required to perform all of the tasks that a Visual Component or Content Component Creator can perform, functionality from the Visual Object and Content Object Tools is available in the Assembly Tool. A Visual Object menu provides access to functionality to create, search for, or edit a Visual Object. The same functionality is provided for Content Objects via the Content Object menu.

The first step to be performed is the creation of a new Visual Object. The Visual Object panel is provided to the assembler in order to create Visual Objects. The panel is the same as can be seen in the Visual Object Tool in figure 5.3. Once an assembler has filled in the required information, a new Visual Object is created and stored in the Assembly Tool's temporary internal data model. The image from the Visual Object appears on screen in order for Content Objects to be linked to it, this can be seen in figure 5.7.

Content Objects must now be created and linked to elements of the Visual Object. The assembler puts the interface into Content Object creation mode and clicks on an element of the Visual Object. The Content Object panel will appear in order for the assembler to enter the relevant information for a new Content Object. This is the same panel and procedure as discussed previously in relation to the Content Object Tool as seen in figure 5.5. Once all of the information has been provided the Content Object is stored in the internal data model. The mapping between the Visual Object element and the Content Object is also stored. This process is repeated until the assembler has created and linked Content Objects for each element of a Visual Object for which they wish to provide supplementary information.

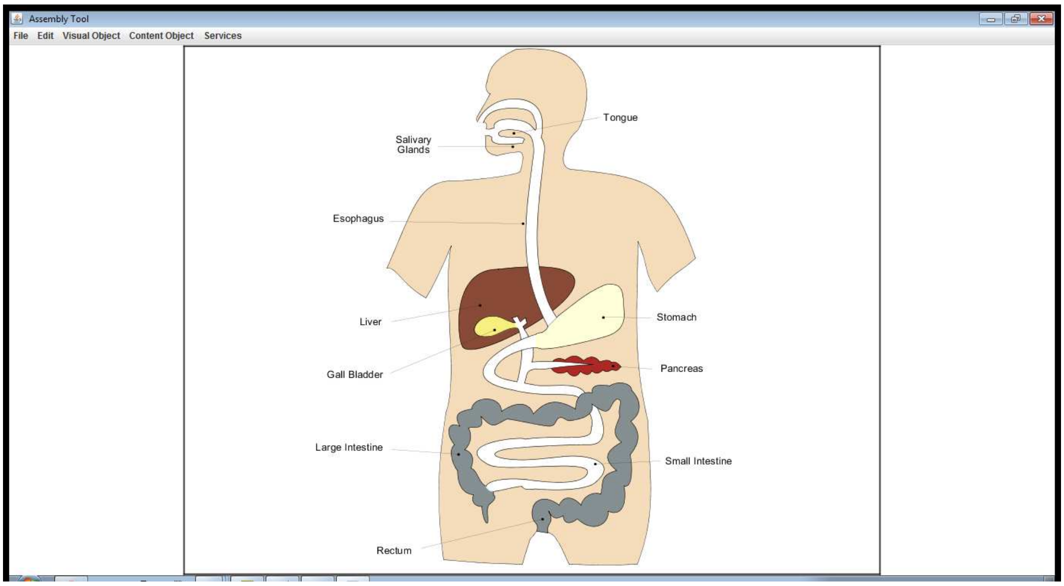


Figure 5.7: Assembly Tool With Image

Metadata is required for the multimodal graphic as a whole and a metadata panel is provided in order for the assembler to provide it. The panel can be seen in figure 5.8 and provides fields for title, subject, description, creator and language, as required in the Assembly File Model. This information is also stored in the internal data model of the interface.

As each Content Object is being created, a sequence is being maintained in the internal data model. The order of the sequence matches the order that each Content Object was added. This sequence is used to define the sequence of Content Objects in the Assembly File Model that can be used to aid learner guidance. In a real world implementation the assembler should be provided with functionality to alter the default sequence. As sequencing is primarily related to the delivery of the multimodal graphic, which is outside the scope of this research, sequencing functionality was not implemented in the proof of concept system as this time. Therefore, the assembler can proceed with saving their multimodal graphic as discussed in section 5.4.3.4.

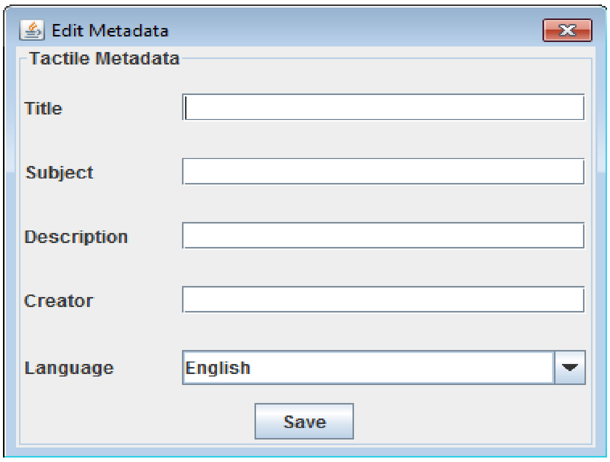


Figure 5.8: Assembly Tool Metadata Panel

5.4.3.3 Complete Component Reuse

A non-expert multimodal graphic assembler can be supported during the assembly process. Functionality is required to search for Visual and Content Objects that can then be combined in order to create a coherent multimodal graphic. The first step to be performed is to locate a Visual Object. The process for locating a Visual Object in the assembly Tool is the same as the first step in editing a Visual Object in the Visual Object tool. The search panel is provided for the assembler to locate a suitable Visual Object, as can be seen in figure 5.4. The assembler can either list every Visual Object or search for a Visual Object using keywords or a known identifier. Once a suitable Visual Object has been identified it can be imported into the Assembly Tool and it's image displayed. The internal model retrieves the Visual Object using the relevant service in order to display it but it does not create a new Visual Object.

The assembler can now locate suitable Content Objects which can be linked to elements of the Visual Object. The interface is placed in Visual Object search mode and the assembler clicks on an element of the Visual Object. The same process is followed for Content Objects as for Visual Objects above. The search panel appears allowing an assembler to search for a suitable Content Object to link to the element of the Visual Object that they clicked. Once a suitable Visual Object is located, it is imported into the Assembly Tool. The Content Object is stored in the internal model and a link to the Visual Object element is maintained. As with the Visual Object a new Content Object is not created. This process is repeated in order to locate suitable Content Objects for each element of the Visual Object to which an assembler wishes to link supplementary information.

Metadata is the only element of the multimodal graphic that did not exist until this point. The metadata panel is used to add metadata for the multimodal graphic the assembler is creating, as can be seen in figure 5.8. The information is stored in the interfaces internal data model. The assembler is now free to save their multimodal graphic as discussed in section 5.4.3.4

5.4.3.4 Storage

When an assembler chooses to store their completed multimodal graphic, they are asked to select an Assembly File Service they wish to use. The services that will be used to store the Visual and Content Objects were set when the information was provided using the Visual Object and Content Object panels. The decomposition engine interacts with the interfaces internal model and performs certain functions based on the type of method an assembler is using. Any objects that were newly created must be sent to the relevant repository for model creation and persistent storage. The objects are sent to the service that was selected in the panel and saved. The identifier that was given to the object is returned to the decomposition engine, which it stores in its internal model. This process is repeated for each newly created object. If an object already existed, the identifier is stored but a new object is not created. This reduces the replication issues from which previous approaches have suffered and has significant benefits in time and cost. Once the Visual and Content Objects have been stored, the engine turns to the creation of an Assembly File. An Assembly File Object is created containing the metadata for the multimodal graphic, the sequence of the Content Objects, the identifiers and locations of each component object and the mappings between the Visual Object elements and the relevant Content Objects. The Assembly File Object is sent to the service that was chosen by the assembler in order for it to be persisted.

5.4.3.5 Editing

If an assembler wishes to edit their multimodal graphic at a later date, they can avail of the edit mode. The assembler is provided with a search panel in order to locate their multimodal graphic's Assembly File, as can be seen in figure 5.9.

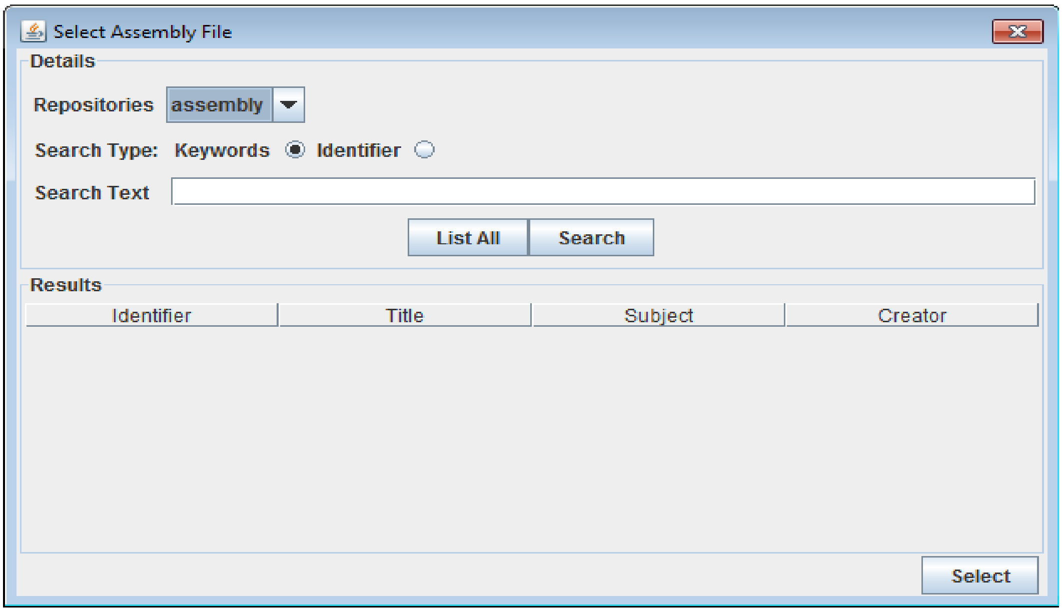


Figure 5.9: Assembly Tool Search Panel

The process of searching for an Assembly File is identical to that of searching for a Content or Visual Object. Every available Assembly File can be listed or a search can be performed using keywords or a known identifier. Once an Assembly File has been located, it is selected and passed to the composition engine. The composition engine retrieves the assembly file object from the relevant service. The interfaces internal data model is populated with the metadata, sequence and mapping information from the object. The location and identifiers of the Visual and Content Objects are used to retrieve the components that are part of this multimodal graphic. Once retrieved the image from the Visual Object appears on screen. An assembler can now use the functionality of the Visual Object and Content

Object menus to alter the contents of their multimodal graphic. If changes are made the relevant objects are updated in their respective repositories. Any alterations are available in real time to anyone who wishes to reuse the objects.

5.4.4 E-Learning Integration

Moodle was chosen as the Virtual Learning Environment into which multimodal graphics could be embedded. That particular VLE was selected as it is the platform that Dublin City University chose for it's E-Learning delivery. The Moodle plug-in works in tandem with the Assembly Tool in order to integrate multimodal graphics into an E-Learning environment. This methodology is similar to that of the Reload tool [RELOAD, 2010]. With the Reload tool course creators can locate and sequence learning objects that exist in learning object repositories. The sequence of learning objects can be packaged together for delivery using the SCORM [ADL, 2004] standard. The SCORM package can then be uploaded into an E-Learning environment, such as Moodle, and delivered to the learner using an embedded player. As no multimodal graphic packaging standard exists, and it's design is beyond the scope of this research, the Moodle interface for MGADF will interact directly with an Assembly File Service and retrieve the content dynamically. This compares with the content retrieval approach of service oriented system like APeLS and KnowledgeTree as discussed in chapter 2. We will assume that assemblers have created a number of multimodal graphics and they are already available in a repository.

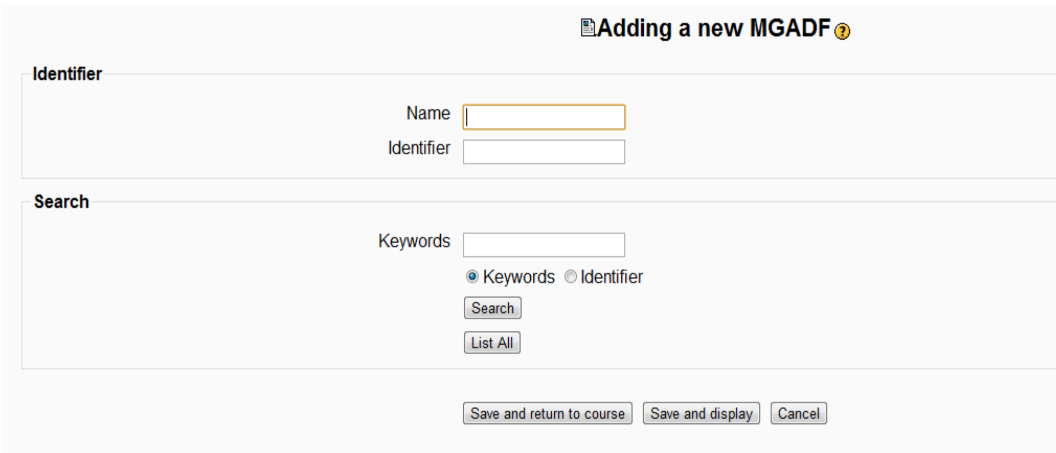


Figure 5.10: Moodle Activity

We will use the term teacher from this point on as it relates to the Moodle role that is capable of adding content to Moodle courses. The first step for a teacher wishing to embed a multimodal graphic into Moodle is to select an Assembly File Object from a given repository. MGADF is available as an activity type for Moodle teachers to select. Figure 5.10 shows the interface that the teacher is provided with once they select the MGADF activity type from the list.

The teacher can either list every Assembly File Object or perform a search using keywords or a known identifier. The search is performed by means of a Web Service query from Moodle to a specific Assembly File Service. The WSDL URI for the Assembly File Service is pre-configured by an administrator and stored in a database table. Once the teacher has located a suitable Assembly File Object they select it. The name and identifier of the selected Assembly File are stored in a database table. A link to the activity is placed on the course content listing for the given week, in compliance with Moodle's interface design. The name of the link corresponds to the title of a given multimodal graphic. In order for a learner to access the multimodal graphic, they click on the link. A Web Service request is sent to the specific Assembly File Service in order to retrieve the Assembly File object with the previously stored identifier. Once retrieved, Web Service requests can be placed to the Visual and Content Object services containing the multimodal graphic components listed in the Assembly File. Once all content has been retrieved, it is displayed on screen for the learner to interact with. As delivery is outside the scope of this research, further details of Moodle delivery are not provided in this thesis.

5.5 Summary

This chapter discussed the implementation of the Multimodal Graphic Assembly and Delivery Framework. The technologies used in the implementation were discussed in section 5.2 and their role in the architecture illustrated. The methods exposed by the Web Services for the interfaces to use were described in section 5.3 as was their configuration and repository functionality. The use of the proof of concept interfaces was illustrated in section 5.4 with the aid of screenshots. Interfaces for Visual Component Creators, Content Component Creators and Multimodal Graphic Assemblers were discussed in terms of the methodologies described in section 4.4. The integration of multimodal graphics into the Moodle Virtual Learning Environment was also described.

Chapter 6

Evaluation

6.1 Introduction

The seventh objective of the research was to, "Evaluate the system with non-expert multimodal graphic assemblers to assess the benefits of the approach". This chapter discusses the evaluation of the Multimodal Graphic Assembly and Delivery Framework (MGADF) using a variety of methods and metrics. The framework consists of three core components; the System Architecture, the Information Architecture and the methodology. Each component is evaluated individually in order to assess the capabilities of the framework as a whole.

The primary research question defined in chapter 1 was, "Can mainstream E-Learning concepts relating to componentisation, information architecture and system architecture, be applied in an approach to multimodal graphic assembly"? Two sub questions were outlined; is the approach feasible and does it provide benefits to the assembler of multimodal graphics? Seven goals were identified which stated that the approach in this research should;

1. Facilitate simple creation of multimodal graphics by non-experts in multimodal graphic assembly.
2. Facilitate faster creation of multimodal graphics when exploiting reusability.
3. Be capable of being integrated into mainstream E-Learning platforms.
4. Maintain a separation between graphical components such that they are independently reusable.

5. Facilitate the assembly and delivery of multimodal graphics and their components in multiple environments.
6. Be capable of supporting various types of visual and non visual content combinations.
7. Be extensible in order to support additional functionality.

The extent to which the research has answered the research question and achieved it's goals is demonstrated in this chapter. Section 6.2 discusses an end user evaluation, which consists of task analysis and user survey evaluation methods. The performance and usability of the approach is assessed through performance based and self-reported metrics. In section 6.3 the data models are compared with systems from the state of the art in accessible graphics to assess their completeness. A use case scenario is used to illustrate the data models ability to represent various visual and non-visual content formats. A scenario based architecture evaluation method is used in section 6.4 to investigate the extensibility of the System Architecture. The chapter ends in section 6.5 with a discussion of the evaluation results and how they relate to the research question and it's numerous goals.

6.2 Methodology

A usability evaluation took place in Dublin City University in October 2010. The evaluation was primarily concerned with the methodology element of the MGADF. It was designed to assess the extent to which the research had met it's first four goals. In addition, the feasibility and benefits of the approach were investigated along with elements of the primary research question. As a proof of concept system had been successfully developed it was clear that the implementation of the approach was feasible. What remained was to assess whether the approach was feasible for the end user and if so what benefits or problems it provided.

6.2.1 Trial Evaluation

An evaluation protocol was designed in order to assess the core elements of the research. Four core elements of the approach were identified for end user evaluation. The first element, was the users ability to assemble a multimodal graphic by creating all of their own content. This would mimic the approaches used in systems such as IVEO [ViewPlus, 2010a]and T3 [TouchGraphics, 2010a] as discussed in section 3.7. The second element, was the users ability to assemble a multimodal graphic by reusing existing content that resided in a repository. As reusability forms the core of the approaches methodology, this task would assess it's impact and provide data for comparison against the state of the art. The third element, was a combination of the first two areas, the ability for a user to reuse existing content whilst also creating new content. This scenario might occur if a user wished to reuse an existing image but provide their own supplementary information. The final element was to assess the users ability to integrate their completed graphics into a mainstream E-Learning environment.

The protocol was trialled with four evaluation participants. The trial served to identify issues with the protocol design that could be altered prior to performing the evaluation with formal participants. The trial participants had similar skillsets, backgrounds and age ranges as the formal evaluation participants. A combination of quantitative data, using task analysis and qualitative data, using a questionnaire and subjective workload analysis, were selected for the evaluation. A within-subjects repeated measures approach was taken, involving the same group of participants performing numerous tasks.

The evaluation began with the participant signing a consent form. They were then provided with a period of instruction where all of the functionality of the Assembly Tool, which they would be required to use in the evaluation tasks, was demonstrated. The participants assembled two multimodal graphics during this phase of the protocol. Before beginning the tasks, a period of instruction for the Moodle plug-in was provided. The participants integrated one multimodal graphic into Moodle during the period of instruction.

Once instruction was completed the participants were asked to perform 5 tasks. Task 1 required the participant to assemble a multimodal graphic of the "Human Digestive System" by creating a Visual Object based on an available SVG file and creating Content Objects by typing in their own content. Task 2 required the participant to assemble a multimodal graphic of "The Human Eye" by reusing a Visual Object and a number of Content Objects from a repository. Task 3 involved the assembly of a multimodal graphic of "The Human Tooth" by reusing a Visual Object from the repository and typing in their own content. Task

4 involved the assembly of a multimodal graphic of "The Human Respiratory System" by creating a Visual Object based on an SVG file and reusing a number of Content Objects from the repository. The final task involved integrating a randomly selected multimodal graphic, created during one of the previous four tasks, into the Moodle E-Learning environment.

Participants were asked to follow a "think-aloud" protocol by verbalising what they were thinking and doing as they performed each task. This would provide some insight into the users level of understanding of the process and highlight any cognitive issues that led to negative results. Participants were asked to assess their workload after each task, using a NASA Task Load Index (TLX) [Hart and Staveland, 1988] [Moroney et al., 1992] workload assessment sheet. Performance metrics for task success and task time were noted for each task. Each participant filled in an online questionnaire when all tasks were complete.

The trial uncovered a number of problems with the design of the evaluation. Firstly, participants struggled to grasp the ability to "think-aloud" whilst performing the tasks. Some participants were narrating every step they made but not why they were performing those steps, others were not saying anything at all. Additionally, it has been shown that if you are interested in task time, a "think-aloud" approach can negatively impact the speed of a task [Tullis and Albert, 2008]. If a participant is spending too much time describing their actions or engaging the evaluator in conversation, their timing results can be skewed. For these reasons it was decided to remove the "think-aloud" instruction from the protocol used in the final evaluation.

It was discovered that the tutorial phase of the protocol was taking a long time to complete and participants were being exposed to too much of the interface. As the evaluation was a within-subjects design, task order was to be randomised in an effort to counterbalance any order effects. The tutorial phase however was exposing participants to the functionality of both no reuse and reuse before the tasks began. Therefore, the approach to participant instruction was altered. For the formal evaluation, a tutorial of the functionality required for a given task would be provided prior to that task being performed. Providing the tutorial in this manner ensured that a participant was exposed to the functionality for the tasks in the same order the tasks are to be performed.

In order to support simple localisation, it has been argued in chapter 4 that no labels should be placed on the images that are to be used as Visual Objects. If labels are required by the user they should be displayed dynamically during delivery or when printing a tactile copy. During the trial, images without labels were used to simulate this approach. However, as the participants were not necessarily domain experts in the topics being used, they were provided with a printed copy of the image with labels attached for reference purposes. In order to attach content to specific regions of the graphic, the participant had to find the region on the printed graphic, match that location to the on screen graphic and then search for or enter the relevant content for that region of the graphic. This added to the cognitive load of the participant, led to some confusion and impacted on task time. As the presence of labels on the on screen graphic would not interfere with the underlying approach being evaluated, it was decided to provide labelled onscreen images as a guide for the formal evaluation participants.

If a user is creating new Content or Visual Objects, they must provide metadata in order for the information to be subsequently searched for and reused. During the trial, participants were free to enter their own metadata. This had an impact on task completion times as people paused in order to decide on adequate metadata to enter for the objects. As this introduced a variance into the tasks, it was decided that sample metadata would be provided in order to ensure that each participant would enter the same information.

Initially, participants were not able to complete the evaluation in the hour allocated to it. Tasks 3 and 4 were essentially a combination of the functionality contained in Tasks 1 and 2 and provided little additional insight into the feasibility and benefits of the approach. Therefore, the evaluation protocol was reduced to three tasks; the assembly of a multimodal graphic without the support of reuse, the assembly of a multimodal graphic with the support of reuse and the integration of a multimodal graphic into a mainstream E-Learning environment. The final protocol is discussed in the next section.

6.2.2 Formal Evaluation

The formal evaluation was completed by 22 participants. The participants were recruited through various mailing lists and all volunteered to take part. There were 13 males and 9 females aged between 22 and 51. Numerous backgrounds were represented in the group;

academics, postgraduate students, administrative staff and assistive technology professionals. The majority considered themselves non-experts in multimodal graphic assembly. Each participant received an honorarium of 20 Euro for taking part in the evaluation. Evaluation sessions lasted approximately 40 minutes.

6.2.2.1 Tasks

The evaluation consisted of three tasks. Task 1 required participants to assemble a multimodal graphic of the "Human Digestive System" without the support of reusable content. There were 4 steps to complete. In step 1, participants created a new Visual Object by importing an SVG file representing the "Human Digestive System", that was stored on the computer, into the Assembly Tool and adding descriptive metadata. In step 2, participants created three Content Objects and linked them to regions of the graphic. The regions were; "Liver", "Stomach" and "Large Intestine". Participants entered a title, 2 content fragments and some descriptive metadata for each of the three Content Objects. In step 3, participants entered metadata for the overall multimodal graphic. The final step was to save the multimodal graphic to the repository. Participants were provided with a document listing the steps they had to perform and containing any sample content that they were required to enter. The procedure, image and sample content for the task can be seen in appendix B.4.

Task 2 required participants to assemble a multimodal graphic of the "Human Eye" with the support of reusable content. There were 4 steps to complete. In step 1, participants performed a keyword search of a repository in order to locate an existing Visual Object that contained an image of the "Human Eye". Once located they selected it for use in their multimodal graphic. In step 2, participants searched for a number of suitable Content Objects that could be linked to three regions of the graphic. The regions were; "Cornea", "Lens" and "Vitreous Humour". A keyword search was performed in order to find suitable content which was then imported and linked to the relevant regions. In step 3, participants had to enter metadata for the overall multimodal graphic. The final step was to save the multimodal graphic to the repository. Participants were provided with a document listing the steps they had to perform and containing any sample content that they were required to enter. The procedure, image and sample content for the task can be seen in appendix B.5.

Task 3 involved integrating a multimodal graphic, that had previously been created in either Task 1 or Task2, into a sample online course using the Moodle E-Learning environment. Participants clicked a link in the Moodle interface in order to open a window that allowed them to run a keyword search for the required multimodal graphic. From the list of results, participants selected the correct graphic. They then clicked a save button and a link to the selected multimodal graphic was placed in the Moodle course. As the procedure was short, and period of instruction was provided directly before it, the participant was informed verbally of which multimodal graphic they were required to integrate.

6.2.3 Instruction

A period of instruction, containing only information relevant to a specific task, was performed directly before each task was carried out. Prior to Task 1, participants were shown how to create a new Visual Object, how to create numerous Content Objects, how to enter metadata for the entire multimodal graphic and how to save the multimodal graphic. The sample multimodal graphic created during the tutorial related to "‘Permanent Teeth"’. The procedure, image and sample content for the tutorial can be seen in appendix B.1.

Prior to Task 2, participants were shown how to search for a Visual Object, how to search for numerous Content Objects, how to enter metadata for the entire multimodal graphic and how to save the multimodal graphic. The sample multimodal graphic created during the tutorial related to a "‘Nerve Cell"’. The procedure, image and sample content for the tutorial can be seen in appendix B.2.

Prior to Task 3, participants were shown how to open the search window in the Moodle interface, how to search for a multimodal graphic in the repository and how to save it to the course listing. The sample multimodal graphic that was integrated into the Moodle course during the tutorial was "‘Permanent Teeth"’. The procedure, image and sample content for the tutorial can be seen in appendix B.3.

6.2.3.1 Protocol

Prior to beginning the evaluation each participant signed a consent form. They were asked to consent to having their audio and computer interaction recorded using screen capturing software [TechSmith, 2010]. Once consent was given the recording began. The evaluator provided the participant with a background to the research where they saw a comparison between a mainstream graphic and a tactile graphic. A demonstration of a multimodal graphic was also provided. Once the participant was ready to begin they performed the three tasks. Task time and task success performance metrics were noted for each task. As this was to be a within-subjects study, skewing of the results due to order effects was a danger. In order to avoid this, a counter balancing technique, involving the rotation of Tasks 1 and 2 for each participant, was carried out. Every second participant therefore performed Task 2 prior to performing Task 1. Different graphics and content of equal difficulty were used for Tasks 1 and 2 so as to reduce the influence of learnability. In addition, the multimodal graphic that the participant was required to search for in Task 3 was altered for each participant. This resulted in four combinations of task order and integrated multimodal graphic, namely;

- Task 1, Task 2, Task 3 (The Human Digestive System)
- Task 2, Task 1, Task 3 (The Human Eye)
- Task 1, Task 2, Task 3 (The Human Eye)
- Task 2, Task 1, Task 3 (The Human Digestive System)

After each task, participants were asked to indicate their workload for the task using the NASA Task Load Index. Participants received a sheet containing six rating scales; Mental Demand, Physical Demand, Temporal Demand, Performance, Effort and Frustration. They were asked to place a mark on each scale that represented their workload for that rating for the given task. Marks lower on the scale indicated a lower workload than those higher on the scale. There were 21 gradients on the scale each representing a workload level of 5 allowing for a range of 0 to 100. An example of the form used can be seen in appendix B.6.

Once all tasks had been completed participants were asked to fill in an online questionnaire. The questionnaire contained 16 questions consisting of personal information, Likert scales and open ended questions. Once the questionnaire was completed participants were provided with an honorarium of 20 Euro and signed a form to confirm they had received it.

6.2.4 Results

In the following section quantitative performance metrics are described relating to task time and task success. Qualitative self reported metrics are described relating to the NASA TLX and the questionnaire.

6.2.4.1 Task Time

The mean task times for participants to complete the tasks, as shown in table 6.1, were; 424.23 seconds for Task 1, 140.91 seconds for Task 2 and 19.09 seconds for Task 3. These times suggest that assembling a multimodal graphic with the support of reusable content is on average 67% faster than assembly without that support. In addition, taking 19.09 seconds to integrate a multimodal graphic into a mainstream E-Learning course would not place a high temporal demand on an instructor.

Time On Task (Seconds)			
Tasks	Task 1	Task 2	Task 3
Mean	424.23	140.91	19.09

Table 6.1: Time On Task

A paired samples t-test was conducted to compare the speed of assembling a multimodal graphic with no reusability support with the speed of assembly with reusability support. There was a significant difference in the scores for no reuse (M=424.23, SD=78.56) and reuse (M=140.91, SD=30.96) conditions; $t(21) = 19.58$, $p=0.000$. These results suggest that time benefits can be gained by multimodal graphic assemblers if they are supported with repositories of reusable content.

6.2.5 Task Success

The mean task success for all three tasks was 100% as shown in table 6.2. The primary concerns of the evaluation were the speed differences of the tasks, the end users opinion of the approach and the ability for a non-expert to successfully assemble multimodal graphics. Learnability of the proof of concept interface was not under investigation and therefore each participant was following a printed procedure in order to complete each task. The participant was allowed to ask the evaluator questions at any time if instructions were confusing.

It should be noted that although the procedure was available for participants to follow, the possibility was there for a participant to be incapable of completing it. During the evaluation some participants made errors including; selecting incorrect menus, attaching content objects to the wrong location and running incorrect keyword searches. Although we were not noting error rates during the evaluation, they were observed and could have lead to a lack of task completion. As the assembly of a multimodal graphic was the primary factor in determining success, and as each participant who made errors managed to recover successfully to complete the task, the success rate stands at 100%.

It must also be considered that in a real world scenario for Task 1, the assembler would be required to draw a suitable graphical image, or have it drawn for them in order to assemble a multimodal graphic. As the graphical image and all necessary content was provided to each participant this reduced the complexity of the task. Given the pre task tutorial, the perceived intuitiveness of the interface, as indicated by self reported metrics and the relative simplicity of the tasks, this success rate is not surprising. Due to the factors outlined above, no significance is claimed in relation to task success and the results are included for illustration only.

Task Success (%)			
Tasks	Task 1	Task 2	Task 3
Mean	100	100	100

Table 6.2: Task Success

6.2.6 Workload

A modified version of the NASA Task Load Index was used during the evaluation. The process suggested by NASA involves a combination of weights and ratings [NASA, 2010]. There are six possible rating headings; Mental Demand, Physical Demand, Temporal Demand, Performance, Effort and Frustration. For a full explanation of each scale see appendix B.7. The first step suggested by NASA is to show the participant 15 cards consisting of a combination of two of the rating headings. The participant should select which of the two rating headings most contributed to workload for the task they completed. In the second step, the participant should be asked to place a mark on a scale for each heading indicating the degree to which that heading impacted on workload for the task they completed. The final TLX score in then computed using a combination of the weights and the ratings.

Research into the approach indicated that the weighting element of the TLX confused participants and added additional time onto the completion of the TLX process [Nygren, 1991]. In addition, it showed that a modified version of the TLX, where only the rating scales were used, provided similar results to the approach combining weights with the ratings [Beyers et al., 1989] [Moroney et al., 1992]. Given this fact, and that a modified version of the TLX has been used in previous assistive technology evaluations [Stevens and Edwards, 1996], the modified approach was used in this evaluation. For each task, the users selections for each rating were summed and an average computed. The final results, which can be seen in table 6.3 show a mean TLX score of 19.77 for Task 1, 15.83 for Task 2 and 8.06 for Task 3. A low workload score for Moodle integration suggests that participants did not find the process overly demanding.

Overall Workload (TLX Score)			
Tasks	Task 1	Task 2	Task 3
Mean	19.77	15.83	8.06

Table 6.3: Overall Workload

A paired samples t-test was conducted to compare the workload of assembling a multimodal graphic with no reusability support with the workload of assembly with reusability support. There was a significant difference in the scores for no reuse (M=19.77, SD=11.92) and reuse (M=15.83, SD=10.10) conditions; $t(21) = 2.329$, $p=0.030$. These results suggest that a multimodal graphic assemblers workload is lower when they are supported with repositories of reusable content.

Although the overall workload scores are significantly lower when reusability is being harnessed, additional insights can be gained by examining the results for each individual rating. The results for each rating scale can be seen in table 6.4.

There was a significant difference in the scores for mental demand for no reuse (M=24.09, SD=15.85) and reuse (M=17.97, SD=13.42) conditions; $t(21) = 2.667$, $p=0.014$. These results suggest that a multimodal graphic assemblers mental demand is lower when they are supported with repositories of reusable content. We must bear in mind that the repositories used in the evaluation contained a small amount of data and therefore limited the number of search results. If a user had to preview a large number of results in order to locate suitable content an increased mental demand would be expected.

There was a significant difference in the scores for physical demand for no reuse (M=18.18, SD=16.44) and reuse (M=12.50, SD=10.32) conditions; t(21) = 2.907, p=0.008. These results suggest that a multimodal graphic assemblers physical demand is lower when they are supported with repositories of reusable content. This result was expected as an assembler is required to perform additional tasks if they need to import an image and provide all of the information for Content Objects as opposed to searching for suitable Visual and Content Objects. As instructors have previously avoided the use of accessible graphics due to the amount of work involved [Sheppard and Aldrich, 2001] these results are promising for future uptake.

There was a significant difference in the scores for temporal demand for no reuse (M=24.31, SD=16.24) and reuse (M=17.27, SD=12.88) conditions; t(21) = 2.980, p=0.007. These results suggest that a multimodal graphic assemblers temporal demand is lower when they are supported with repositories of reusable content. As the time associated with the creation of accessible graphics was quoted as a reason instructors usually avoid them [Sheppard and Aldrich, 2001], a quicker approach may lead to more widespread use of multimodal graphics.

There was no significant difference in the scores for performance for no reuse (M=12.50, SD=8.55) and reuse (M=11.13, SD=9.99) conditions; t(21) = 0.781, p=0.444. As participants achieved a 100% success rate in all 3 tasks it is not surprising that no significant difference is found in performance between Tasks 1 and 2.

There was no significant difference in the scores for effort for no reuse (M=22.27, SD=18.69) and reuse (M=21.13, SD=16.96) conditions; t(21) = 0.313, p=0.758. This result is surprising given a difference in task time of 67% and the significant differences in mental, physical and temporal demand. Qualitative feedback would suggest that due to the pre task instructions the participants received and the perceived ease of use of the interface, they did not believe they exerted any considerable effort in order to complete either task.

There was no significant difference in the scores for frustration for no reuse (M=17.27, SD=15.84) and reuse (M=15.00, SD=14.05) conditions; $t(21) = 1.156$, $p=0.261$. These results are not surprising given the participants could ask questions at any time and were provided with a documented procedure for each task. Any frustration was predominantly down to flaws in the user interface design and was common to both tasks. It must also be considered that participants only assembled a single multimodal graphic in both tasks. Had they been asked to assemble numerous multimodal graphics using each approach it may have altered the results.

Individual Ratings (TLX Score)			
Scales	Task 1	Task 2	Task 3
Mental Demand	24.09	17.95	9.09
Physical Demand	18.18	12.50	6.59
Temporal Demand	24.31	17.27	9.09
Performance	12.50	11.13	6.13
Effort	22.27	21.13	8.86
Frustration	17.27	15.00	9.31

Table 6.4: Individual Rating Scales

6.2.6.1 Questionnaire

Each participant completed a post evaluation questionnaire consisting of 16 questions to provide qualitative feedback. The questions consisted of personal information, such as age and sex, a number of Likert scales on various aspects of the approach and some open ended questions for general feedback. Space was provided for participants to provide optional additional information for the majority of questions. The questionnaire can be seen in appendix B.8. The results of the questionnaire are provided in the following sections.

In question 4 participants were asked, “How would you describe your level of expertise with mainstream E-Learning?”. The scale of available answers ranged from Very Inexperienced to Very Experienced. 2 participants chose Very Inexperienced, 4 chose Inexperienced, 6 chose Neutral, 8 chose Experienced and 2 chose Very Experienced, as can be seen in figure 6.1. The results show that the majority of the group either had no opinion or tended towards the experienced end of the scale. This would suggest a level of familiarity with mainstream E-Learning amongst the participants.

In question 5 participants were asked, “How would you describe your level of expertise with multimodal graphic assembly?". The scale of available answers ranged from Very Inexperienced to Very Experienced. 14 participants chose Very Inexperienced, 4 chose Inexperienced, 4 chose Neutral and 1 chose Experienced, as can be seen in figure 6.2. It is clear from the results that the participants consisted of primarily inexperienced multimodal graphic assemblers, a group whom this research is aimed at assisting.

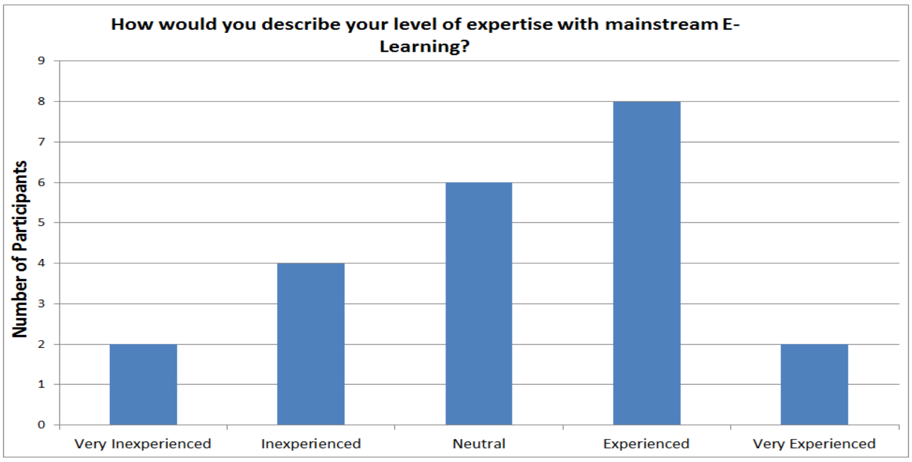


Figure 6.1: Question 4

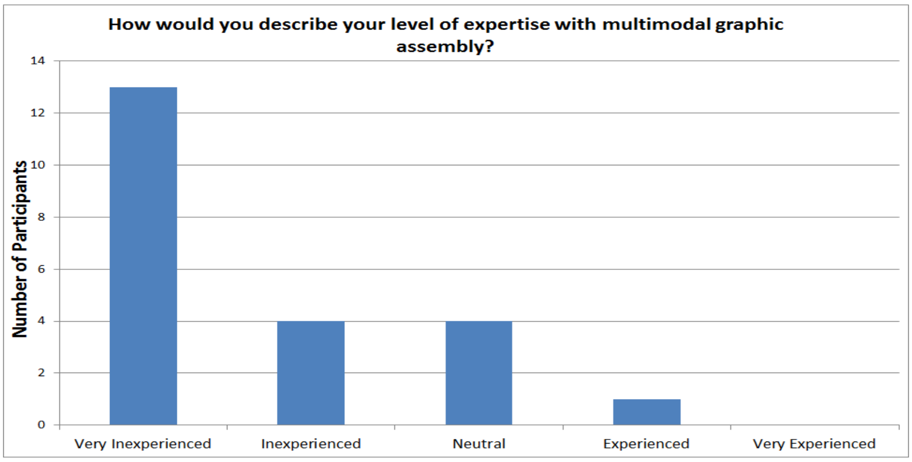


Figure 6.2: Question 5

In question 6 participants were asked, "Have you ever been required to teach a graphical concept to a Blind learner?" 4 participants chose Yes and 18 chose No, as can be seen in figure 6.3. This reinforces the result from the previous question that participants were not familiar with accessible graphics as they had not been required to use them previously. If Yes was chosen, a follow up question of "How did you illustrate the graphical concept to the Blind learners?" was asked. Of the 4 participants that chose Yes, none of them had harnessed multimodal graphics. Their answers were; "By describing verbally what the visual graphic was", "Could only provide a textual description of graphs and trees", "Through verbal descriptions", "Using the idea of shapes they were familiar with, whether it be through an actual object they had experienced or felt, or a shape in a concept that they were familiar with.". The additional comments show verbal description being predominantly used as an alternative to graphical images.

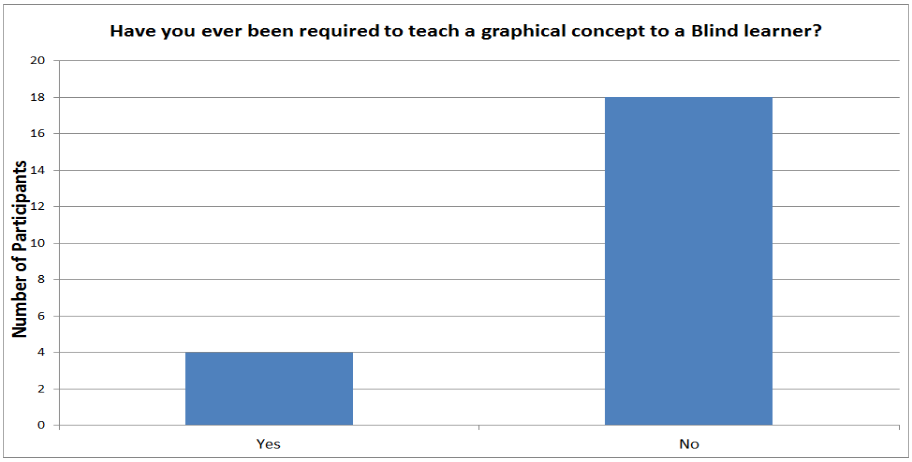


Figure 6.3: Question 6

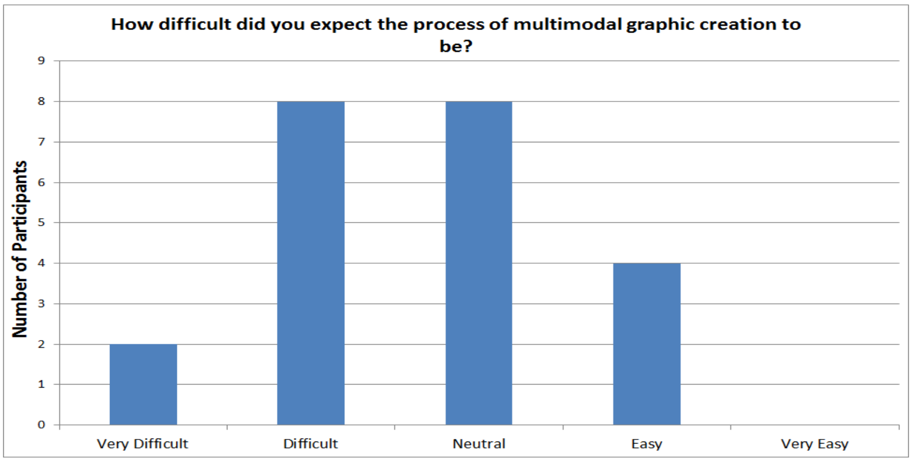


Figure 6.4: Question 8

In question 8 participants were asked, "How difficult did you expect the process of multimodal graphic creation to be?". The scale of available answers ranged from Very Difficult to Very Easy. 2 participants chose Very Difficult, 8 chose Neutral and 4 chose Easy, as can be seen in figure 6.4. The majority of the group either had no preconceptions about multimodal graphic assembly or tended towards the difficult end of the scale. This compares to the common perception in the state of the art that accessible graphics are difficult and time consuming to create [Sheppard and Aldrich, 2001].

In question 9 participants were asked, "How difficult was it to create a multimodal graphic with the assembly tool?". The scale of available answers ranged from Very Difficult to Very Easy. 10 participants chose Easy and 12 chose Very Easy as can be seen in figure 6.5. All participants tended towards the easy end of the scale. This is in contrast to the perception the participants had prior to participating in the evaluation.

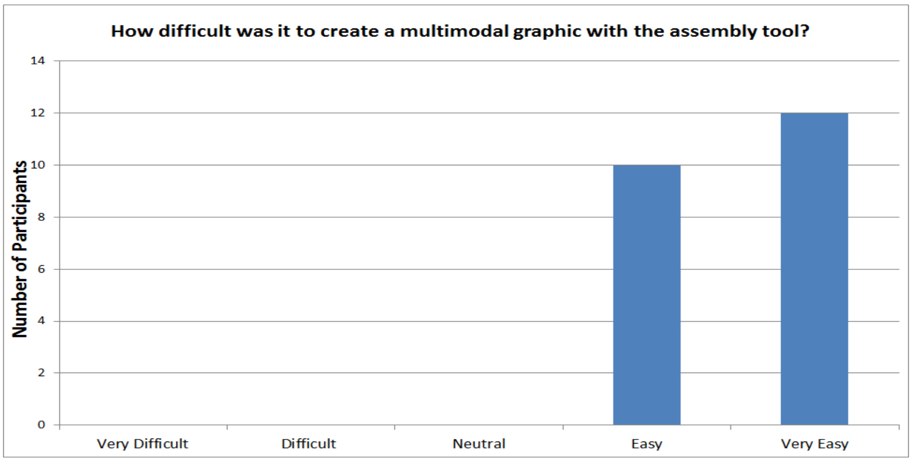


Figure 6.5: Question 9

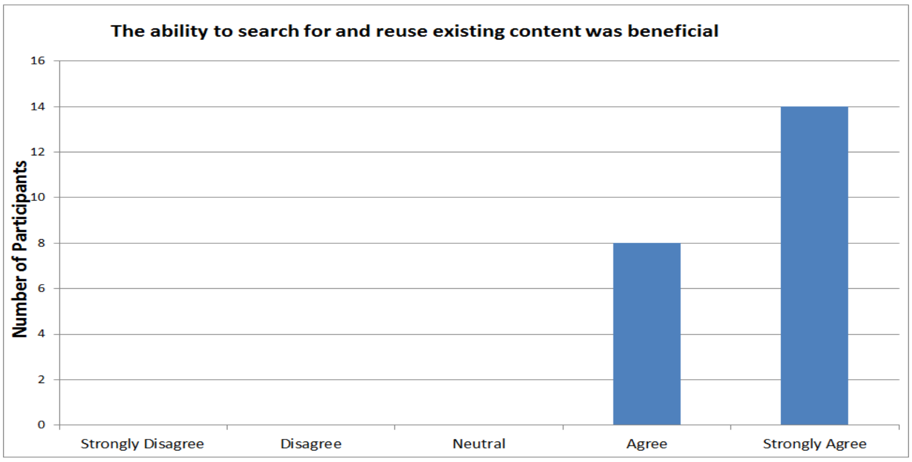


Figure 6.6: Question 10

In question 10 participants were asked their opinion of the statement, "The ability to search for and reuse existing content was beneficial". The scale of available answers ranged from Strongly Disagree to Strongly Agree. 8 participants chose Agree and 14 chose Strongly Agree, as can be seen in figure 6.6. All participants tended towards the agree end of the scale. It is significant that the participants found the support of the reusability beneficial as it correlates with the performance metrics and workload scores discussed previously.

In question 11, participants were asked their opinion of the statement, "I would prefer to create my own content than to search for suitable existing content". The scale of available answers range from Strongly Disagree to Strongly Agree. 2 participants chose Strongly Disagree, 6 chose Disagree, 9 chose Neutral, 3 chose Agree and 2 chose Strongly Agree, as can be seen in figure 6.7.

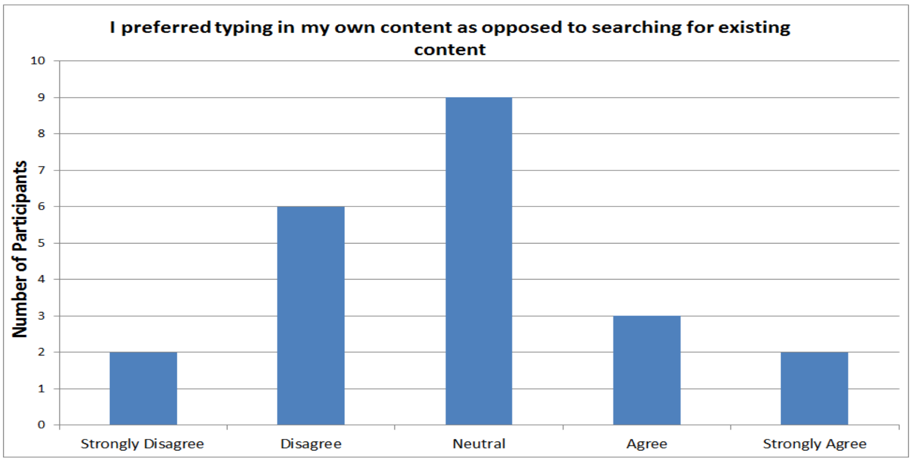


Figure 6.7: Question 11

From discussions with participants it seems that this question was confusing. Some participants took it to mean that they liked the ability to create their own multimodal graphics using the tool as opposed to relying on their creation by others. Other participants understood that the question was intended to ascertain a preference between the lack of reusability support and its existence. This suggests that differences in interpretation led to the fragmentation of the results. It should be noted that the tasks designed to assess limited reuse were removed from the final evaluation protocol. The MGADF approach can cater for an assembler who wishes to reuse existing Visual Objects but provide their own Content Objects. As this question only deals with the extremes of reuse versus no reuse, the middle ground of half reuse was not specifically assessed. The fragmentation however may indicate end user interest in such an approach.

In question 12, participants were asked their opinion of the statement, "I believe the metadata is expressive enough to facilitate searching". All 22 participants chose Yes as can be seen in figure 6.8. Although the participants were not exposed to a wide range of scenarios that would put the available metadata under stress, none of them suggested a scenario where the available metadata was weak. All where able to search for and reuse existing content using the metadata available to them. It should be noted that the participants opinions are based solely on the metadata elements they provided during the evaluation tasks and does not represent a thorough evaluation of the metadata. A more detailed evaluation of the metadata takes place in sections 6.3 and 6.4.

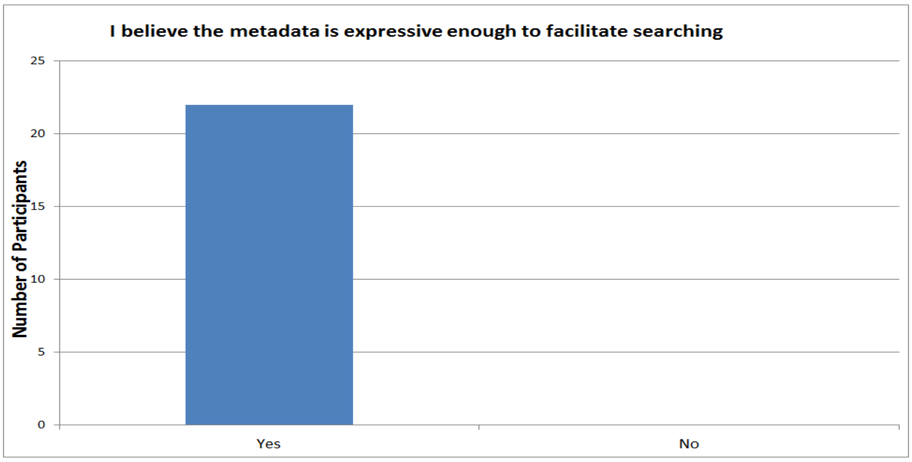


Figure 6.8: Question 12

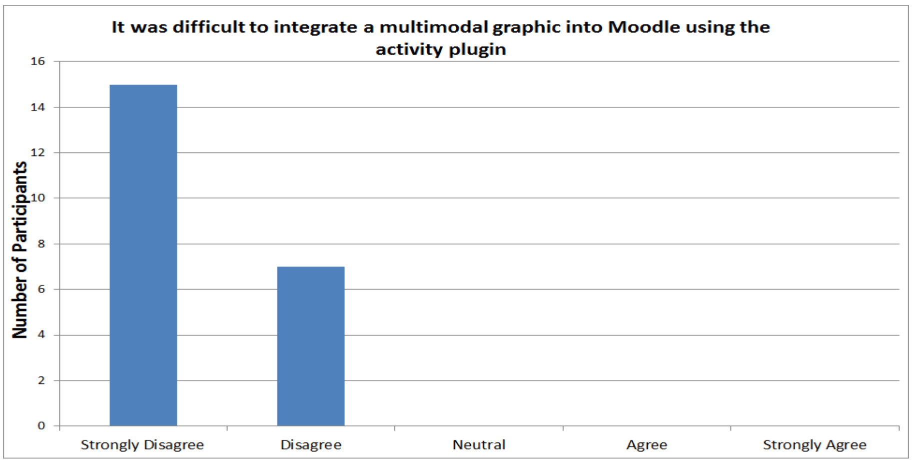


Figure 6.9: Question 13

In question 13, participants were asked the opinion of the statement, "It was difficult to integrate a multimodal graphic into Moodle using the activity plugin". The scale of available answers ranged from Strongly Disagree to Strongly Agree. 15 participants chose Strongly Disagree and 7 chose Disagree, as can be seen in figure 6.9 It can be seen the majority tended towards the disagree end of the scale. This result is significant as it correlates to the performance metric and workload scores for Task 3 and suggests that participants found it easy to integrate multimodal graphics into the Moodle E-Learning environment,

In question 14, participants were asked their opinion of the statement, "I would use the system if I was required to provide graphics to Blind learners". The available answers ranged from Strongly Disagree to Strongly Agree. 5 participants chose Agree and 17 chose Strongly Agree, as can be seen in figure 6.10. The result indicates a positive view from the participants that they would use the system if it were available to them. This is significant as it suggests a strong level of user acceptance for the approach.

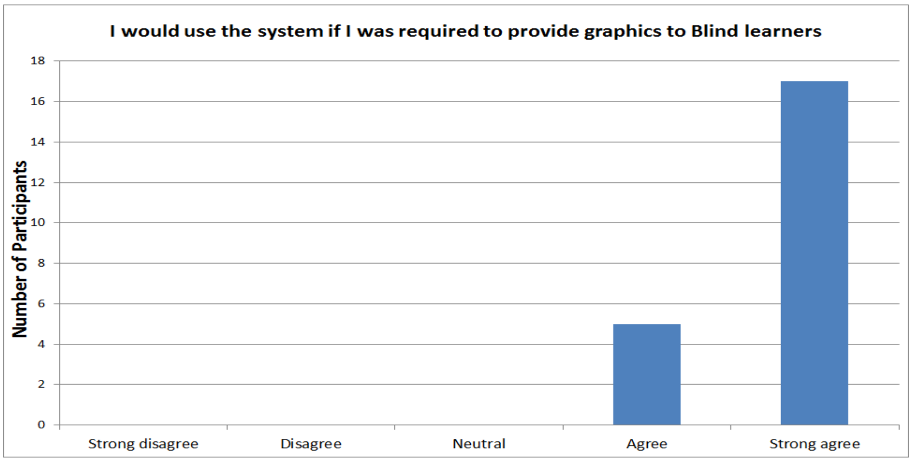


Figure 6.10: Question 14

6.2.7 Perceived Weaknesses

The end user evaluation was designed to evaluate the approach and not the proof of concept Assembly Tool itself. However, during the course of the evaluation usability issues were identified by the participants that had an impact on the user experience and thus their opinion of the approach and ability to complete the tasks. Some are especially relevant to the feasibility of the approach and are discussed in the following sections.

Participants noted that the metadata contained a lot of repetition, specifically the subject and creator fields. This was particularly noted during Task 1 when the participants had to add metadata for a Visual Object, a number of Content Objects and the multimodal graphic as a whole. A level of frustration was evident each time a participant re-entered an element of metadata that had previously been provided. Given the approach relies on content creators populating repositories with content for others to reuse, a repetitive task may reduce their willingness to do so. Participants suggested they would prefer if they were able to enter metadata at the start of the assembly process that could then be used to pre-populate certain fields later in the procedure. This would tie in with the comments of other participants who said that the assembly process should be wrapped in a wizard that can guide the assembler through the steps they needed to perform in order to assemble a multimodal graphic.

Over reliance on mouse clicks was highlighted by a number of participants. For example, when a participant clicked on the "Add Fragment" button in the Content Object Panel, focus did not move to the textbox where the fragment was to be typed. Participants had to click on the textbox prior to typing in each fragment. This process needed to be repeated for each fragment. Additionally, if the participant did not click on the textbox and began to type the fragment, each key would trigger the "Add Fragment" button once again resulting in the creation of numerous fragment text boxes. This behaviour had an impact on task time and the frustration level of the participant. Another area where this became apparent was during the performance of a keyword search. Participants hit the return key after entering their keywords but doing so did not trigger the search. The participant was required to physically click on the "Search" button with their mouse. This frustrated some participants as it did not mimic the behaviour of tools they use everyday. A lack of shortcut keys for primary functionality was highlighted by other participants further emphasising the participants desire for the tool to have the same interaction strategies available as common tools.

A lack of visual feedback was identified in a number of areas. There was no form of feedback to inform a participant of which Content Object mode they were in. Whether the participant selected "New Content Object" or "Search for Content Object" the interface looked exactly the same. The difference between the modes was not apparent until the participant clicked on a region of the graphic and saw either the Content Object Panel or the Search Panel. This affected task times due to participants pausing in an attempt to identify which mode the system was currently in.

Some participants indicated that they would like to have been provided with a visual indication of which elements of the graphic were clickable, for example, if the border of the region altered colour as the mouse moved over it. This was highlighted most when participants attempted to attach a Content Object to the “Vitreous Humour” in the “Human Eye” graphic. The region encompassed a large area and participants where unsure where to click in order to attach the Content Object to the correct region. Participants also suggested that once content had been attached to a region, visual feedback should be provided in order to identify which regions content had been linked to. In the proof of concept interface, an assembler is not provided with any visual feedback to identify which regions of a graphic have Content Objects linked to them and which do not. A user will only be informed that a Content Object is already attached to a region if they try to add another one to the same region. During the evaluation this lack of feedback led to participants being unsure whether they had completed certain steps of the procedure correctly.

Although the Assembly Tools window was maximised by default, panels that appeared for specific functionality were not. Due to this fact one particular problem appeared for some participants. Those that performed Task 1 prior to performing Task 2 had created a multimodal graphic of the “Human Digestive System”. During Task 2 they were asked to search the repository for a graphic of the “Human Eye”. As the keyword “Human” is in the metadata for both graphics, two results appeared. This in itself is not a problem however the panel was too small to tell the difference between the two graphics from metadata alone, as can be seen in figure 6.11. This impacted on task time as participants alternated between the two results in order to locate the Human Eye. Some participants even initially chose the “Human Digestive System” because they had previously created it, even though it obviously did not contain a graphic of the “Human Eye”. The method used to display the results of the keyword search has a direct impact on the feasibility of the approach.

Some participants suggested that a spell checker be added to the interface in order to ensure that titles, fragments and metadata were accurately entered. A flaw in spelling during content creation would have a direct impact on the ability to locate that content using a keyword search. This highlighted how mistakes in data entry could have an impact on the usability of the approach.

One final issue occurred which has implications for localisation. One participant entered their real name into the creator field during the tutorial phase for a task. Their name contained an accented character, and although the content saved correctly it could not be retrieved. Not only could that content not be retrieved but all repository interaction began to return errors resulting in an unusable system. As the interface was not processing special characters correctly, it corrupted the data models in the repository and therefore rendered the approach unusable. It highlighted the need for interfaces that interact with the repositories to handle special characters adequately in order to avoid data model corruption.

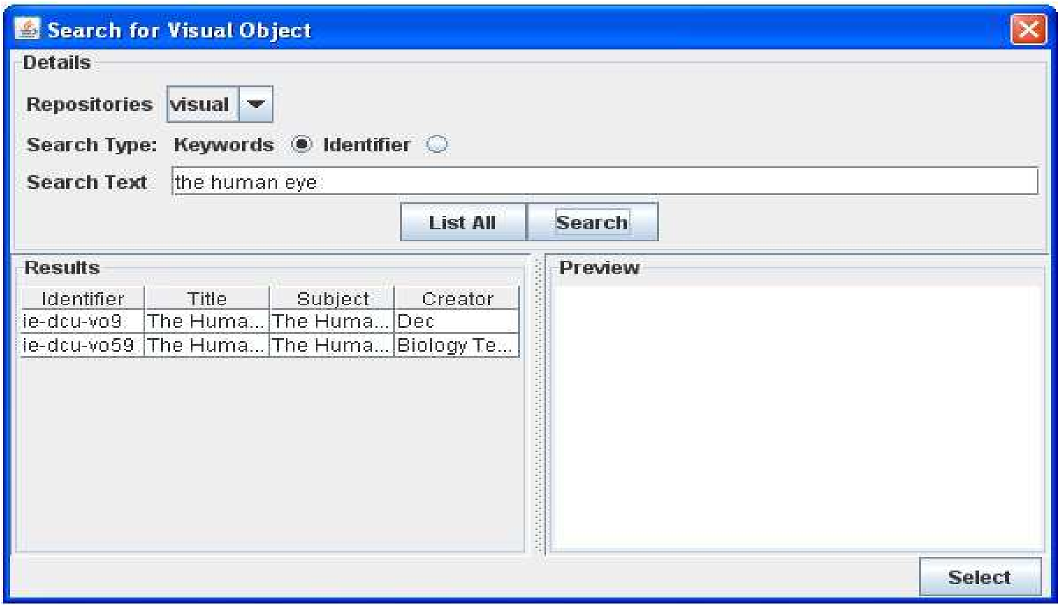


Figure 6.11: Search Problem

It can be seen that the issues discussed above relate predominantly to interface usability. The Assembly Tool was a proof of concept interface that contained all of the necessary functionality to interact with the framework. As it was the approach that was to be assessed and not the interface, the system was deemed fit for purpose as an evaluation tool. Given the issues described above, it is clear however that the usability of the interface has a direct impact on the feasibility and user experience of the approach. As the framework is deliberately abstracted from any specific user interface, each user interface designed to interact with the framework could suffer from its own usability issues. The possible variation in interface design and it's impact on end user experience could be seen as a weakness. In order to improve the user experience of the Assembly Tool, Schneiderman's 8 golden rules [Shneiderman, 1998] should be retrospectively applied. It should be noted that this would not alter the MGADF in any way but would improve the usability of the proof of concept interface.

6.3 Information Architecture

The next area of the MGADF requiring evaluation is it's Information Architecture. Therefore, the completeness of the data models in the framework are evaluated in the following sections. We define completeness as the models ability to represent all of the information required for the concept it represents, be it a Visual Object, Content Object or Assembly File. In order to evaluate the models they are compared to the related work from the state of the art.

It must first be considered whether models exist for each component of a multimodal graphic. An investigation of related work identified three core components of a multimodal graphic; the visual image on which the graphic is based, the supplementary information linked to regions of the image and descriptive metadata that can provide a learner with additional information such as the title, subject and description of a multimodal graphic [McMullen and Fitzpatrick, 2010a]. The three models in the framework are capable of representing data for the three components identified in the state of the art. The existence of a further component varies from implementation to implementation. If an approach does not use pre-annotated image formats such as SVG, it is necessary for the creator to manually annotate regions of a graphic in order for them to be linked to relevant content. In order to facilitate this functionality, an annotation model must be available. As the approach in this research recommends the use of pre-annnotated image formats, an annotation model is not required and therefore we can say that all necessary data models are available.

6.3.1 Completeness

Even though the relevant data models are available we must assess the extent of their completeness. We do this by comparing the frameworks data models with the data models from the related work. T3 uses a proprietary data model based on Macromedia Director [Macromedia, 2010]. As the data model representation in these files can not be inspected by a third party, we cannot compare their data model attributes with those in our models. Instead we will investigate the functionality that the T3 data models support and assess whether our models can support the same functionality.

T3 [TouchGraphics, 2010a] does not use pre-annotated images in it's approach. Because of this the digital version of the image, to which the supplementary content is linked and the tactile version of the image, with which the user interacts, do not need to be visually similar. Multimodal graphic creators use the tactile graphic, and the touchscreen in order to create digital annotations for regions of the graphic. It is the digital annotations that are stored in the model and displayed on screen. As our approach uses pre-annotated image formats, in particular SVG [W3C, 2010], the annotations and the image are located in the same file. The `<content>` tag in our Visual Object Model points to the location of a pre-annotated SVG image file and therefore can represent graphical annotations.

T3 supports two types of supplementary content, text and audio. Textual content is delivered to the user by means of a text to speech engine, and audio content is played using a multimedia player. The `<content>` tag in our Content Object Model can contain plain text, or can point to the location of an audio file. The `<content>` tag is a child of the `<fragment>` tag which contains a "mime-type" attribute that can be used to differentiate between different fragment formats. Therefore, it can represent the same types of supplementary content as T3.

A link is maintained between graphical annotations and supplementary content in the T3 system. This allows for relevant content to be delivered to the learner when they press on regions of the graphic. The `<map>` tag in the Assembly File Model is capable of representing a link between a region of a graphical image and a particular piece of supplementary content, thus mimicking T3 capabilities. Finally, when a learner loads a multimodal graphic using T3 they are provided with background information for the graphic they selected. This information generally contains a title and a description of the overall graphic. The `<metadata>` tag in the Assembly File Model can contain numerous child tags which can be used to represent the same descriptive information as those in T3. Tags are available for title, subject, description and numerous other types of descriptive metadata.

IVEO [ViewPlus, 2010a] uses the SVG image format as a basis for it's data model. As SVG is an open standard we can compare the contents of the IVEO model with the data models of the MGADF in a more specific manner. IVEO takes a different approach to T3 in a number of areas. It uses a pre-annotated image format which does not require the creator to annotate regions of a graphical image. Additionally, the image displayed on screen is the same image that is represented in a tactile form. As we have already discussed, the `<content>` tag of the MGADF Visual Object Model can point to the location of an SVG image file, thereby mimicking both the capabilities and graphical data model of IVEO.

A combination of SVG `<desc>` and `<a>` tags are used by IVEO to provide supplementary content to a learner. IVEO can deliver plain text, audio files, documents and hyperlinks to a learner. The `<content>` and `<fragment>` tags of the Content Object Model have already been discussed in relation to T3. By providing different "mime-type" attributes to the `<fragment>` tag, all four types of content can be represented thus supporting the same range of content capabilities as IVEO.

In order to represent a link between annotated regions and supplementary content, IVEO relies on the various shape types of the SVG standard. Shapes such as `<rect>`, `<polygon>` and `<circle>` are used to depict elements of a graphic and provide annotation. Each shape can take a unique "id" attribute and can contain child `<a>` and `<desc>` tags. When a learner presses on a shape, it's "id" is used to deliver the contents of the corresponding `<desc>` or `<a>` tag for that shape. The `<map>` tag of the Assembly File Model, discussed previously, performs the same function in the MGADF. This allows a region of the graphic to be linked to supplementary content and matches the capabilities of IVEO. IVEO also provides learners with background information relating to the multimodal graphic they have loaded. The abilities of the `<metadata>` tag in the Assembly File Model were already discussed in relation to T3 and therefore can also be used to represent IVEO's background information.

It can be seen that the models in the MGADF can represent every component and every element of core functionality that was identified in the related work. Given this fact, we deem the MGADF data models to be complete. If further investigation should reveal weaknesses in the data models, they can be easily extended due to their use of XML [W3C, 2008].

One final point in relation to metadata must be made. Each MGADF data model uses an element set from the Dublin Core Metadata standard [DCMI, 2004] as child elements of their `<metadata>` tag. Dublin Core consists of 15 metadata elements. Although not all 15 elements are required in each of the MGADF models, the models have been designed to support all 15 elements. Therefore, the MGADF data models contain a complete implementation of the Dublin Core Metadata Standard.

6.3.2 Content Combinations

Although the data models are complete in terms of supporting the same functionality as the state of the art, the sixth goal of the research stated that the approach should "Be capable of supporting various types of visual and non visual content combinations". The aim of the goal was to future proof the approach such that the contribution was not restricted to the domain of touchscreen based audio tactile graphics. As touchscreen based audio tactiles with text to speech, were used as a proof of concept in this research, we will now investigate the frameworks ability to support another form of graphical presentation and interaction.

In the following use case scenario, a researcher wishes to harness the MGADF in order to provide blind learners with access to graphics using haptic interaction. No tactile or touchscreen will be used in this scenario. Haptics can be used to deliver three dimensional concepts in a manner tactiles can not. The recommendation for the use of pre-annotated image formats must be followed in order to exploit the MGADF. The researcher chooses the X3D format [Web3D, 2009] which allows for three dimensional graphics to be represented using an XML syntax. In order for the blind user to interact with the graphic, haptic effects must be added to it. This can be achieved through the use of the H3DAPI library [H3DAPI, 2010], which provides a syntax for embedding haptic effects directly into X3D graphics using XML. The `<content>` tag of the Visual Object Model can point to the location of a haptic enabled X3D file and the `<dc:format>` tag can store the Visual Objects

MIME type, in this case "model/x3d+xml". The researcher does not need to alter the Visual Object Model in order to support the haptic enabled 3D graphic.

The researcher would like the learner to be provided with supplementary information relating to the parts of the graphic they may be interacting with. They have decided that audio files will be used as a content format. The `<content>` tag of the Visual Object Model can be used to point to the location of suitable audio files. The "mime-type" attribute of the `<fragment>` tag can be used to represent the audio format being used for example "audio/mpeg" for MP3 or "audio/x-wav" for WAV. The researcher does not need to alter the Content Object Model in order to support audio content.

A link must be stored between regions of the 3D graphic and relevant supplementary information. Each X3D shape node can contain a unique identifier. The `<map>` tag in the Assembly File Model can be used in order to link an X3D node to a Content Object. For example, `<map coId=''ie-dcu-co1'' voElementId=''node1''/>`, would link the Content Object containing identifier "ie-dcu-co1" with the X3D node containing identifier "node1". The researcher does not need to alter the Assembly File Model in order to create the link.

We have illustrated that the data models can support all of the requirements in this use case scenario. In order for the haptic graphic to be delivered to the learner the developer would need to implement a suitable interface containing a composition engine. The engine would retrieve the components of the haptic graphic, render them on screen and provide interaction capabilities using a haptic device such as a Novint Falcon [Novint, 2010] or Sensable Phantom Omni [Sensable, 2010b]. Each component of the haptic graphic in the MGADF is independently reusable and thus non-experts could be supported in assembling haptic graphics.

6.4 System Architecture

The final goal of the approach was that it should "Be extensible in order to support additional functionality". As the data models are written in XML we know that that they can be extended with minimal effort if required. This section will therefore focus on the extensibility quality attribute of the MGADF's architecture. It should be noted that a Service Oriented Architecture was designed and implemented for the MGADF. The Software Engineering Institute in Carnegie Mellon wrote a report examining the relationship between quality attributes and Service Oriented Architectures [SEI, 2005]. The report assigned a status for a Service Oriented Architectures impact on various quality attributes. A status of green was given to extensibility to indicate that there are "known SOA solutions based on relatively mature standards and technology". As the MGADF approach utilises a Service Oriented Architecture, it should be capable of supporting extensibility.

In order to assess the extensibility of our architecture, a scenario based evaluation is performed. This approach has been shown to work well in architecture evaluation methods such as ALMA [Bengtsson et al., 2004], ATAM [Kazman et al., 1998] and SAAM [Kazman et al., 1994]. The approach used is based on ALMA and requires the completion of five steps [Babar and Gorton, 2004].

Set the Analysis Goal For the purpose of this evaluation we wish to access the extensibility of the MGADF.

Describe the software architecture(s) The architecture of the MGADF was described in detail in section 4.2.

Elicit change-scenarios We envisage a scenario where personalisation support is to be added to the MGADF. It is described in section 6.4.1.

Evaluate the change-scenarios The impact of the alteration on the data model and system components of the MGADF is illustrated in section 6.4.2.

Interpret results The result of the evaluation is outlined in section 6.4.3.

6.4.1 Scenario

As the approach is based on techniques from mainstream E-Learning, it is reasonable to assume that further functionality from that domain will be required in multimodal graphics in the future. The Visualisation component of Siqueiras generic E-Learning architecture

[Siqueira et al., 2003], discussed in chapter 2, contained three elements, Interface, Navigation and Personalisation. To date we have provided proof of concept systems that satisfy the interface requirement, and sequencing capabilities in the Assembly File Model that satisfy the Navigation requirement. Therefore, in this scenario we will consider the remaining area, personalisation. Personalisation was discussed in detail in chapter 2. We will focus on a scenario where suitable content is dynamically selected for delivery to a learner, that matches their preferences and abilities. Froschl identified that a user model can contain domain specific and domain independent data that can be harnessed to provide personalised content delivery [Froschl, 2005]. The axes of personalisation that we will use are the learners preferred language as domain independent data and their content difficulty level as domain specific data.

6.4.2 Evaluation

We will assess a scenario where personalisation functionality is to be added into the MGADF architecture, described in chapter 4. We will begin by investigating the existing services ability to cater for the extension. We have stated in the thesis that Visual Objects should not contain any language specific content in order to enhance their reusability. We will assume for the purpose of this scenario that multiple versions of Visual Objects with various levels of difficulty do not exist. Taking these facts into account, the Visual Object Service will not be impacted in this scenario.

Content Objects can exist in multiple languages and it is possible for the fragments within them to be written for various difficulty levels. For example, if elements of the Human Eye are being described, simpler language and less fragments could be used for a beginner compared to difficult terminology and numerous detailed fragments for an advanced learner. Therefore, the Content Object Services ability to handle these variations must be assessed. As the Content Object Model contains the `<dc:language>` tag, localisation is not a problem. Difficulty level on the other hand was not considered when the Model was designed. The model does however support all 15 Dublin Core elements and one which could be harnessed to support the functionality is `<dc:type>`. The recommended best practice is to use a controlled vocabulary, such as the DCMI Type Vocabulary

[DCMI, 2010a], for the values of this element. This vocabulary contains types such as "Image", "InteractiveResource" and "Sound". For this scenario we will use a vocabulary of difficulty levels that contain the types, "'Beginner", "Intermediate" and "Advanced". If the relevant difficulty type was used as a value for the `<dc:type>` tag, levels of difficulty could be defined in Content Objects. The search functionality of the Content Object Service could then be extended to support searches that include locales and difficulty levels.

The final area of existing functionality to be investigated is the Assembly File Service. Assemblers would need to create multimodal graphics consisting of various language and difficulty level combinations in order for a personalisation engine to choose a suitable multimodal graphic for a specific learner. Therefore, the Assembly File Model must be capable of representing locale information and levels of difficulty. As it utilises the same `<dc:language>` and `<dc:type>` tags as the Content Object Model, the same capabilities are available as those discussed above. If Assembly File Models contained a value in `<dc:language>` and a difficulty level in `<dc:type>`, the search functionality of the Assembly File Service could be extended to support a search including language and difficulty preferences.

We have seen that existing services can support the additional functionality, however we do not have a service capable of managing the preferences and abilities of the learner. Therefore, we must investigate the addition of a User Model Service into the MGADF. As the services are loosely coupled, this should not be a problem. A User Model would need to be designed and implemented capable of representing the language preferences and ability levels of the learner. This model could be placed in a repository and accessed via a User Model Service. The service would need functionality to create, update, search for and retrieve User Models. The design and implementation of such a service would be no different to the design and implementation of the existing MGADF services.

In order for the new functionality to be harnessed, the methodology must be extended to support it. As all composition logic resides in the user interface, a personalisation engine would be required in any interface wishing to harness the additional functionality. This engine could act as a pre-requisite for the composition engine. The methodology would operate as follows. A learner, whose User Model resides in an available repository, interacts with the interface. The learner informs the environment that they wish to load content for the "Human Eye". The learners User Model is queried using the User Model Service in order to retrieve the learners language preference and difficulty level. Using this information the personalisation engine runs a query using the Assembly File Service in order to locate a "Human Eye" multimodal graphic to suit the learners language preference and difficulty level. The identifier of the Assembly File can then be passed to the Composition Engine and the relevant content retrieved and displayed to the learner.

6.4.3 Results

The scenario above represents a simplified view of personalised delivery. In practise a more sophisticated adaptive engine and user model would be required in order to provide the full range of adaptive capabilities found in the state of the art. Existing MGADF models may have to be extended in order to support the axes of personalisation required for a truly adaptive experience. The scenario does however illustrate that personalisation functionality is possible with minimal impact on the MGADF and therefore the approach is deemed to be extensible.

6.5 Discussion

The results of the end user evaluation provide some insights into the extent to which the research question has been answered and it's goals achieved.

6.5.1 Goals

The first goal of the research was for the approach to, "Facilitate simple creation of multimodal graphics by non-experts in multimodal graphic assembly". Non- experts are supported in the creation of multimodal graphics through the ability to search for existing graphical components and combine them into a multimodal graphic. The assembler does not require a skillset in tactile image design and can follow similar construction methodologies as mainstream E-Learning content creators. The majority of evaluation participants were non-experts, as illustrated in the results of question 5 in the questionnaire. Most participants felt that multimodal graphic creation was easy, as illustrated in the results of question 9 in the questionnaire. These results indicate that the research has achieved it's first goal.

The second goal of the research was for the approach to, "Facilitate faster creation of multimodal graphics when exploiting reusability". Previous approaches required assemblers to supply their own visual and content components during the assembly of multimodal graphics. Furthermore, it was not easy to reuse existing content during the assembly of new multimodal graphics. The approach in this thesis allows assemblers to locate existing content and use it in the assembly of their multimodal graphics. Evaluation participants were asked to assemble multimodal graphics in two ways, creating all of the content and reusing existing content. When reuse was being exploited, the completion time for the task was 67% faster than when it was not. Therefore, the research has successfully achieved it's second goal.

The third goal of the research was for the approach to, "Be capable of being integrated into mainstream E-Learning platforms". The use of open data standards in the frameworks Information Architecture provides syntactic interoperability. Additionally, the use of standard Web Service communication protocols allows any interface to interact with the frameworks content. In Task 3, participants integrated multimodal graphics into the Moodle E-Learning environment. The responses to question 13 of the questionnaire indicate that they found it easy to perform the task. These results indicate that the research successfully achieved it's third goal.

The fourth goal of the research was for the approach to, "Maintain a separation between graphical components such that they are independently reusable". The framework employs a component based approach to multimodal graphics in which the visual and content elements of a graphic are maintained separately from one another. This allows for each component to be used independently of one another. In Task 2 participants assembled a multimodal graphic by reusing existing content. Visual and Content Objects were searched for individually and combined in order to assemble a coherent multimodal graphic. The objects had been stored in the repositories prior to the evaluation by content creators. As each component was stored independently of the other and they could be searched for and reused separately, the research has successfully achieved it's fourth goal.

The fifth goal of the research was for the approach to, "Facilitate the assembly and delivery of multimodal graphics and their components in multiple environments". The models in the Information Architecture are based on XML. As XML is an interoperable data standard, the models can be easily parsed and understood by multiple environments. In addition, the System Architecture employs a service oriented design. Web Services are implemented using standard communication protocols. Therefore, any interface capable of communicating using those protocols can interact with the framework in this research. In Task 3 participants used the Moodle E-Learning environment to search for a multimodal graphic and integrate it into a Moodle course. As the graphic had been created previously using the desktop Assembly Tool and it could then be accessed using Moodle the task illustrated the interoperability of the content between the two interfaces. It should be noted that any E-Learning environment capable of communicating via the communication protocols used in the MGADF approach would be capable of interacting with the MGADF data. Therefore, the research has successfully achieved it's fifth goal.

The sixth goal of the research was for the approach to, "Be capable of supporting various types of visual and non visual content combinations". The data models were designed in order to represent various forms of visual and content components. A MIME type is used in the Visual Object Model in order to inform an interface of the type of image on which the object is based. In our implementation this was an SVG file but other graphical file formats are easily supported. MIME types are also used for each fragment in a Content Object. Text only fragments are used in our implementation but it is possible to mix text with audio, document, hyperlink or even video fragments. In section 6.3, the MGADF data models were assessed for their completeness by comparing them to the state of the art. It was illustrated that the use of MIME types in the models facilitated all of the same content combinations that existed in related systems. In addition, a use case scenario was used to evaluate the approaches ability to represent other types of visual and non visual presentation, namely haptic graphics with auditory information. The use case successfully illustrated that the MGADF was capable of supporting such a scenario. As such the research has successfully achieved it's sixth goal.

The seventh and final goal of the research was for the approach to, "Be extensible in order to support additional functionality”. As the framework is service based, new services can be added in order to provide additional functionality without impacting on the design of the approach. As all data models are based on the XML data standard, they can be easily extended to support additional functionality. In section 6.4 the extensibility of the MGADF System Architecture was assessed using a scenario based evaluation method. The theoretical addition of personalisation functionality was discussed. As the existing services were capable of supporting the addition and a new service was included in the architecture with minimal impact, the research has successfully achieved it’s seventh research goal.

6.5.2 Research Question

The primary research question of this work was, “Can mainstream E-Learning concepts relating to componentisation, information architecture and system architecture, be applied in an approach to multimodal graphic assembly?”. Two sub questions were outlined, namely, is the approach, once designed, feasible and does it provide benefits to the assembler of multimodal graphics?

The development of the proof of concept system illustrated that it was feasible to implement the approach but it remained to be seen if was practical for the assembler to use. The evaluation participants did not struggle to complete the evaluation tasks. The majority of participants said that the approach was easy to use as illustrated in the results of question 9 in the questionnaire. They also indicated that they would use the tool if it were available to them as illustrated in results of question 14 in the questionnaire. Therefore, it has been proven that the approach is feasible for the assembler.

In terms of the benefits, the task time results suggested that it was 67% faster to assemble a multimodal graphic when reusable content was available in comparison to when it was not. Participants responded favourably when asked if the reusability support was beneficial as illustrated in the results of question 10 in the questionnaire. In addition, the workload results for Task 1 where reuse was not harnessed were higher than in Task 2 where it was. This suggests that an approach based on mainstream E-Learning techniques can provide benefits in the area of multimodal graphic assembly.

As it was possible to design and implement an approach based on methodologies, system architectures and information architectures inherent in mainstream Technology Enhanced Learning and given that the evaluation of the approach has produced encouraging results, a positive response can be given for the research question.

6.6 Summary

This chapter discussed the evaluation of the Multimodal Graphic Assembly and Delivery Framework. It began with a review of the primary research question and research goals as set out in chapter 1. Three elements of the framework were identified for individual evaluation. These consisted of the methodology, Information Architecture and System Architecture. Section 6.2 discussed an end user evaluation that took place in Dublin City University designed to assess the frameworks methodology. Results were gathered using a combination of performance and self reported metrics. The results were presented and discussed in relation to the research goals. Section 6.3 outlined an evaluation of the frameworks data models. They were compared with the state of the art in order to assess their completeness. Their ability to represent various combinations of visual and non visual content was evaluated by means of a use case scenario. Section 6.4 investigated the extensibility of the MGADF architecture. A scenario based evaluation approach was used to investigate the feasibility and difficulty of adding personalisation support to the Multimodal Graphic Assembly and Delivery Framework. The chapter ended with section 6.5 which discussed the extent to which the research questions had been answered and the goals achieved.

Chapter 7

Conclusion

7.1 Introduction

This thesis presented research that aimed to apply techniques from mainstream E-Learning to facilitate the assembly of multimodal graphics. An approach was presented called the Multimodal Graphic Assembly and Delivery Framework (MGADF), which consists of a component based Service Oriented Architecture and data models based on open data standards. The approach separates multimodal graphics into independently reusable components and stores them in object repositories. The components in the repositories can be searched for, retrieved and composed into coherent multimodal graphics. XML based data models are used for each individual component. Although each model is independent of the other, they can be combined in order to produce a coherent multimodal graphic. The object repositories are exposed using Web Services. Multiple interfaces can interact with the same services in order to assemble and deliver multimodal graphics. New services can easily be developed in order to offer additional functionality. The framework is capable of supporting non-experts in assembling multimodal graphics and integrating them into mainstream E-Learning environments. Section 7.2 illustrates how the objectives of the research were satisfied and identifies the areas of the thesis that relate directly to each objective. The contribution that this research provides to the area of accessible graphics is provided in section 7.3. The issues that remain to be solved now that the research is complete are highlighted in section 7.4. The chapter concludes in section 7.5 with a discussion of future work.

7.2 Objectives

Seven objectives were identified in order to satisfy the goals of this research. A discussion of each objective takes place in the following sections illustrating how each one was satisfied during the course of the research.

The first objective was to "Research and identify specific techniques in the areas of componentisation, information architecture and system architecture, that can be applied in the area of multimodal graphic assembly.". Mainstream techniques relating to learning objects, learning object repositories, multiple models and Service Oriented Architectures were all applied in the design and implementation of the framework. Details of the techniques were provided in chapter 2 and the solutions employed in the framework, based on those techniques, were discussed in chapter 4.

The second objective was to "Identify the independent components that form a multimodal graphic". In chapter 4 the independent components were identified as the Visual Component, the Content Components and the metadata component. These components formed the basis for repositories, services and object types in the framework.

The third objective was to "Design an extensible architecture capable of supporting the independent components of a multimodal graphic and allowing for their interaction". Section 4.2 discussed the design of the System Architecture in the framework. A Service Oriented Architecture was designed that allows the components of a multimodal graphic to be independently stored and retrieved. A specific object known as an Assembly File is maintained in order to allow the independent components to be combined into a coherent multimodal graphic.

The fourth objective was to "Design data models that can be used to represent the independent components of a multimodal graphic and provide interoperability". Section 4.3 discussed the design of XML Schema Definitions for each component in the framework. The use of XML to represent the data models and the use of the Dublin Core metadata specification provides syntactic interoperability for any interface wishing to interact with the data.

The fifth objective was to "Design a methodology for combining independent components into a coherent multimodal graphic". This was discussed in section 4.4. Functionality should be provided in each assembly interface in order to decompose a multimodal graphic into it's component parts. Once the delivery of a multimodal graphic is requested the independent components of a graphic must be recomposed into a coherent whole. The logic to provide this functionality was defined as the decomposition and composition engines. When a graphic is decomposed into its component parts, an Assembly File is maintained that informs an interface which independent components form a complete graphic. In order to deliver the graphic, the composition engine queries an Assembly File and uses the information within it to retrieve the relevant components and combine them into a single multimodal graphic.

The sixth objective was to "Implement a proof of concept system that clearly demonstrates the feasibility of the approach". Chapter 5 described the implementation of proof of concept interfaces that were designed to demonstrate the feasibility of the framework. Multiple roles were catered for with the implementation of the Visual Object, Content Object and Assembly Tools. The Assembly Tool provides non-experts with the ability to search for existing components and combine them into multimodal graphics. The integration of multimodal graphics into an E-Learning environment was satisfied with the implementation of a plug-in for the Moodle Virtual Learning Environment.

The seventh objective was to "Evaluate the system with non-expert multimodal graphic assemblers to assess the benefits of the approach". Chapter 6 provided details on the evaluation of the framework. The methodologies were assessed by means of an end user evaluation. The Information Architecture was assessed for completeness and a use case scenario used to illustrate it's ability to model numerous visual and non visual content combinations. The extensibility of the System Architecture was assessed using a scenario based architecture evaluation method.

The extent to which the goals of the research were met and the research questions answered was discussed in detail in section 6.5.

7.3 Contribution to State of the Art

This research investigated the application of mainstream E-Learning techniques relating to componentisation, information architecture and system architecture in the area of multimodal graphic assembly. The primary contribution to the area of accessible graphics is the Multimodal Graphic Assembly and Delivery Framework (MGADF). The MGADF is a novel approach to multimodal graphic assembly which successfully applies techniques from mainstream E-Learning in order to provide non-experts with the ability to easily assemble multimodal graphics. To the best of the authors knowledge, mainstream E-Learning techniques have not previously been applied in the area of multimodal graphic assembly and delivery.

The research illustrated the ability for multimodal graphics to be successfully integrated into mainstream E-Learning environments. Previous approaches have operated independently of E-Learning environments thereby separating blind learners from their sighted peers. This approach allows sighted and blind learners to interact with the same E-Learning environment whilst being presented with graphical information.

The MGADF provides a service oriented system architecture, a model based information architecture and methodologies than can be compared with mainstream courseware construction methodologies. These components can be used by future researchers as a basis for investigations into accessible graphics. They can also be integrated into the next generation of commercial multimodal graphic systems. The proof of concept implementations of the MGADF, developed during the research, can be made available immediately, to provide end users with access to the benefits of the approach.

Instructors have often avoided the use of accessible graphics due to the perceived time consuming nature of their creation. The evaluation participants in this research found the MGADF approach easy to use and said that they would use it if it were available to them. The statistics suggest that an assemblers workload is lower when supported with reusability from the MGADF compared to a lack of reusability prevalent in previous approaches. Additionally, it is indicated that the assembly of multimodal graphics can be performed quicker using the approach suggested in the MGADF. Given these facts, if the MGADF was embraced by instructors, it may lead to an increase in the existence of accessible graphics.

The work has been published in seven peer reviewed conferences at both national and international level [McMullen and Fitzpatrick, 2008] [McMullen and Fitzpatrick, 2009] [McMullen, 2008] [Fitzpatrick and McMullen, 2008] [McMullen and Fitzpatrick, 2010b] [McMullen, 2010] [McMullen and Fitzpatrick, 2010a].

7.4 Remaining Issues

This section will discuss the issues that remain now that the period of research is complete. In order for the method of assembler support to work, it requires Visual Component and Content Component Creators to make their content available in object repositories exposed using Web Services. There are numerous institutions around the world who specialise in tactile graphic design. Institutions, like The National Centre for Tactile Diagrams [NCTD, 2010] or The American Printing House for the Blind [APH, 2010], will create a tactile graphic for a client for a fee. It is envisaged that should an institution like this move to a repository based model, they would charge for access to such a resource. Although the framework could support paid access to a service, it would create a restriction on non-expert assemblers that does not exist when searching for graphics suitable for sighted learners. It is also possible for non-expert repositories of images to appear which could be provided on a no fee basis but the quality of the images may not be suitable. The success of the approach relies on the availability of multimodal graphic components and this can be seen as a limitation.

The approach is based on the benefits of reusable components. As the primary component of a multimodal graphic is the image itself, it is paramount that the Visual Components are reusable in multiple contexts. In most situations this is the case, a map of the USA is set and will not change, the same can be said for the “Human Digestive System” or the “Human Eye”. However, let us consider areas of Mathematics where graphics are changing constantly, the hands on a clock, the angles of a triangle or the size of the wedges in a pie chart for example. As the data in those graphics changes, new images are required. It is fair to say that reusability can still be performed in terms of multiple languages. Tactiles may need to be created in multiple languages for each variation of the hands on a clock. With our approach the same image can be reused for each language. However, as an assembler would need to locate a new image for each variation in the hands on the clock face, the overhead required may outweigh the reusability benefits of the approach. Therefore, it can be seen that the approach works best for topics where the layout of the graphics do not change or change rarely.

The architecture used in the framework is a component based Service Oriented Architecture. The service based nature of the approach was put in place in order to expose the content in the repositories to numerous educators and to support collaboration. For example, it should be possible for a Visual Component Creator to operate independently but provide their content for others to use. The service based approach provides this functionality and has proven to be feasible. It has also been shown that an assembler may also play the role of the Content Component and Visual Component Creator, which the architecture supports, but it raises a question. What if the assembler does not wish to make their content available for others to use? The framework can be implemented at multiple levels of granularity. Services may be available at an international level, a national level, an institution level or a local level. It is fair to say that the assembler could install services at a local level that are not accessible by anyone else. They therefore get the benefits of reusing the content they create if they wish to localise it or use it in another context. However, should an assembler be required to set up services if the content will never be accessed by others? A variation of the approach may be required where the component based nature of the framework is maintained but the service oriented nature of the approach removed.

An issue has emerged regarding the context sensitive nature of the components. This should not impact Visual Components as they are by their nature context sensitive. It is more likely to have an impact on Content Components. If we take a Content Component regarding the livers role in the “Human Digestive System” for example. This Content Component can contain numerous different fragments of information regarding the liver. When an assembler is reusing this component they must reuse it in its entirety. The assembler may be creating another multimodal graphic which also contains an element for the human liver but they do not need to provide information on it’s role in the digestive system. It is conceivable that a number of fragments in the liver component, that deal with the liver in general and not its role in the digestive system, would be useful to the assembler at this point. Therefore, the reusability of content components at a fragment level would be beneficial.

The use of pre-annotated graphical formats, such as SVG, was recommended in this research. The recommendation aimed to make the assembly process as intuitive as possible. The ability for an assembler to click on an element of a Visual Object and immediately link it to a Content Object aids in the simplification and speed of the assembly process. It does, however, raise an issue. What if an image is not annotated to a granularity that an assembler requires. For example, let us consider a graphic showing the limbs of the human body. The assembler wishes to provide information on the arm. The image creator may have annotated the arm as a single element. If the assembler wishes to provide information on the upper arm and lower arm separately, they cannot do it. The assembler would be required to open the graphical image in a suitable drawing program and re-annotate the arm into two separate hit testable elements. This could then be imported into the assembly interface and separate content objects linked to each piece of the arm. The assembler may not have the necessary software or skill to re-annotate the image or the image may not be available to them for download. For this reason, future iterations of the approach should consider the reintroduction of annotation functionality for those who wish to fine tune the Visual Objects at their disposal.

Although the framework has been designed to support other forms of multimodal graphic, the implementation in this thesis was focussed on touchscreen based audio tactiles. The primary aim of the research was the integration of multimodal graphics into mainstream E-Learning environments and it was successful. However, an issue remains that could limit it's widespread adoption. Paper based tactiles are static by nature. This approach provides the ability to print a single tactile image and use it to deliver content in multiple contexts. The tactile image itself however is static and can not be altered. If alterations are made to an image it must be reprinted in a suitable tactile form and given to a learner. It is fair to say that most E-Learning material is produced in advance and the relevant tactile could be sent to the learner prior to the beginning of a course, similar to the distance learning content used by the Open University. However, sighted learners do not have to wait for printed versions of graphics used in E-Learning environments. Therefore, a version of the framework should be implemented and evaluated that benefits from a haptic interface or a large refreshable Braille display which can deliver graphics to learners in a truly dynamic fashion.

7.5 Future Work

A lot of research has taken place in the area of Adaptive Educational Hypermedia (AEH) relating to the adaptive delivery of learning content based on the contents of a learners user profile. Section 6.4 assessed the extensibility of the MGADF using a scenario where personalisation was to be added to the framework. As mainstream E-Learning is moving in the direction of personalisation, future work should consider implementing the hypothetical scenario and assessing the benefits of personalisation in the delivery of accessible graphics. Users have shown preferences towards the speed and prosody of their text to speech engine, the language the content is made available in and the type of interface used to access it. In addition, there are various types of instructional strategies and learning styles that may need to be supported in advanced multimodal graphic systems. All of these elements can be catered for in order to provide dynamic personalised multimodal graphic interaction.

Should this work take place, attention should be paid to the standards “Individualized adaptability and accessibility in e-learning, education and training” [ISO, 2008] and “User profile preferences and information” [ETSI, 2010]. These standards provide information models which can be used to represent the needs and preferences of learners who wish to access online content. They also facilitate the description of learning content in a manner that defines it’s suitability for specific learners and interfaces. The combination of both capabilities allows content to be dynamically selected to suit a specific users needs, preferences and user interface.

There are times when a learner may not have access to an Internet connection and therefore would be unable to retrieve content from the component services in real time. A requirement for a multimodal graphic packaging system was outside the scope of the research. Future work could investigate the design of such a packaging system, keeping in mind the ability to support multiple content types, interfaces and interaction methods. The existence of such a packaging standard would increase the portability and interoperability of multimodal graphics. It is envisioned that future iterations of audio tactile systems, such as T3 and IVEO, would use a service based approach for content management as advocated in this thesis. However, as an interim step, an Export Engine could be designed to package the contents of MGADF Visual and Content Objects in a format suitable for delivery on related systems. For example, the content of relevant Content Objects could be embedded into a suitable SVG Visual Object and delivered using the IVEO audio tactile system.

Section 7.4 illustrated how the use of pre-annotated image formats are suggested in this thesis in order to reduce the complexity of the multimodal graphic assembly process. It also highlighted the need to consider the introduction of annotation functionality in future iterations of the approach. This functionality may be required if a user wishes to fine tune an image that is not annotated to their liking, or if they wish to use a graphical format that does not support embedded annotation. For example, if a BMP image format was to be used, the user would require the ability to define important regions of the image by means of an overlay annotation layer. A separate information model would need to be created in order to represent and store the annotation information. A mapping would also need to be maintained that linked an annotation model with the visual component of a specific Visual Object. This would require a minor extension to the Assembly File model currently available. Section 6.4.2 illustrated the extensibility of the MGADF system and information architectures, therefore the inclusion of the annotation functionality described above is possible with minimal impact on the current approach.

As mentioned previously in section 4.3.4, the integrity of the information models was beyond the scope of the research. Should components of an Assembly File be deleted from their repositories, no update is made to the Assembly File. Therefore, a user may select an Assembly File for delivery and be presented with an error that certain components could not be retrieved. In order to combat this, a test suite that can validate the integrity of Assembly Files should be considered. The suite should also be capable of validating whether the contents of a given model are compliant with the schema defined for it.

Solutions for some of the remaining issues could be investigated as part of future work.

A fragment service could be added in order to provide reusable content at a fragment level. This approach would be similar to that proposed by ALOCOM [Verbert and Duval, 2007] [ALOCOM, 2010]. Assemblers could take fragments from multiple Content Objects and combine them into a Content Object for use in a different context. This approach would increase data management requirements on behalf of the services but would provide the assemblers with more customisation during the assembly process.

Other forms of presentation and interaction should be investigated as part of future work. Section 6.3 discussed the ability of the framework to support haptic graphics. The use case should be implemented in order to assess the frameworks ability to support three dimensional graphics and to assess the benefits that a haptic form of interaction may have for blind learners. Additionally, in order to combat the static nature of paper tactiles, an implementation should take place that allows the framework to interact with a refreshable pin matrix display such as that used in the HyperBraille project [Völkel et al., 2008]. The combination of a dynamic display with the framework in this research could provide for truly dynamic multimodal graphic integration in E-Learning environments.

Bibliography

[ADL, 2004] ADL (2004). SCORM. *http://www.adlnet.gov/Technologies/scorm/SCORMSDocuments/2004%204th%20Edition/Overview.aspx*.

[ADL, 2010] ADL (2010). Advanced Distributed Learning. *http://www.adlnet.gov*.

[AHEAD, 2008] AHEAD (2008). Seeing AHEAD: a study of the factors affecting blind and vision impaired students going onto third level. *http://www.ahead.ie/publications_12_seeingahead.php*.

[AHVITED, 2009] AHVITED (2009). AHVITED Project Website. *http://www.ahvited.org*.

[AICC, 2010] AICC (2010). The Aviation Industry CBT Committee. *http://www.aicc.org*.

[Aldrich and Sheppard, 2000] Aldrich, F. and Sheppard, L. (2000). 'Graphicacy' the fourth 'R'? *Primary Science Review*, 64:8–11.

[Aldrich and Sheppard, 2001] Aldrich, F. K. and Sheppard, L. (2001). Tactile graphics in school education: perspectives from pupils. *Bri*, 19:69–73.

[Aldrich et al., 2002] Aldrich, F. K., Sheppard, L., and Hindle, Y. (2002). First steps towards a model of tactile graphicacy. *British Journal of Visual Impairment*, 20:62–67.

[ALOCOM, 2010] ALOCOM (2010). ALOCOM Framework. *http://hmdb.cs.kuleuven.be/alocom/*.

[Anoto, 2010] Anoto (2010). Anoto Website. `http://www.anoto.com/`.

[Apache, 2010a] Apache (2010a). Apache HTTP Server Project. `http://httpd.apache.org/`.

[Apache, 2010b] Apache (2010b). Lucene. `http://lucene.apache.org/java/docs/index.html`.

[APH, 1997] APH (1997). Guidelines for Design of Tactile Graphics. `http://www.aph.org/edresearch/guides.htm`.

[APH, 2010] APH (2010). American Printing House for the Blind. `http://www.aph.org/`.

[ARIADNE, 2010] ARIADNE (2010). ARIADNE Foundation Website. `http://www.ariadne-eu.org/`.

[Avizzano et al., 2003] Avizzano, C., Marcheschi, S., Angerilli, M., Fontana, M., Bergamasco, M., Gutierrez, T., and Mannegeis, M. (2003). A Multi-Finger Haptic Interface for Visually Impaired People. *Robot and Human Interactive Communication*, pages 165–170.

[Babar and Gorton, 2004] Babar, M. A. and Gorton, I. (2004). Comparison of Scenario-Based Software Architecture Evaluation Methods. In *11TH ASIA-PACIFIC SOFTWARE ENGINEERING CONFERENCE, (2004) 600*, page 607.

[BANA, 1983] BANA (1983). Guidelines for Mathematical Diagrams. Technical report, American Printing House for the Blind.

[BANA, 2010] BANA (2010). Braille Authority of North America. `http://www.brailleauthority.org/`.

[Bengtsson et al., 2004] Bengtsson, P., Lassing, N., Bosch, J., and van Vliet, H. (2004). Architecture-level modifiability analysis (ALMA). *J. Syst. Softw.*, 69:129–147.

[Beyers et al., 1989] Beyers, J. C., Bittner, A., and Hills, S. G. (1989). *Advances in Industrial Ergonomics and Safety*, chapter Traditional and raw task load index (TLX) correlations: are paired comparisons necessary?, pages 481–485. Taylor and Francis.

[Blackboard, 2010] Blackboard (2010). Blackboard Website. `http://www.blackboard.com/`.

[Bouras and Tsiatsos, 2006] Bouras, C. and Tsiatsos, T. (2006). Educational Virtual Environments: Design Rationale and Architecture. *Multimedia Tools and Applications*, 29:153–173.

[Brady et al., 2005] Brady, A., Conlan, O., and Wade, V. (2005). Towards the Dynamic Personalized Selection and Creation of Learning Objects. In Richards, G., editor, *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2005*, pages 1903–1909, E-Learn 2005–World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education. AACE.

[Brady et al., 2006] Brady, A., O'Keeffe, I., Conlan, O., and Wade, V. (2006). Just-in-time Generation of Pedagogically Sound, Context Sensitive Personalized Learning Experiences . *International Journal on E-Learning*, 5(1):113–127.

[Brusilovsky, 1996] Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6:87–129.

[Brusilovsky, 2004a] Brusilovsky, P. (2004a). Adaptive Navigation Support: From Adaptive Hypermedia to the Adaptive Web and Beyond. *PsychNology Journal*, 2:7–23.

[Brusilovsky, 2004b] Brusilovsky, P. (2004b). KnowledgeTree: A Distributed Architecture for Adaptive E-Learning. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, WWW Alt. '04, pages 104–113, New York, NY, USA. ACM.

[Brusilovsky et al., 2008] Brusilovsky, P., Wade, V. P., and Conlan, O. (2008). *Architecture Solutions for E-Learning Systems*, chapter From Learning Objects to Adaptive Content Services for E- Learning, pages 243–261. IGI Global.

[CBA, 2003] CBA (2003). Canadian Braille Authority Report on Tactile Graphics. *http://www.canadianbrailleauthority.ca/docs/Report_Tactile_Graphics_part3.pdf*.

[CBA, 2010] CBA (2010). Canadian Braille Authority. *http://www.canadianbrailleauthority.ca/*.

[Conklin, 1987] Conklin, J. (1987). Hypertext: An Introduction and Survey. *Computer*, 20:17–41.

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

[Conlan et al., 2002] Conlan, O., Dagger, D., and Wade, V. (2002). Towards a Standards-based Approach to e-Learning Personalization using Reusable Learning Objects . In Driscoll, M. and Reeves, T. C., editors, *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2002*, pages 210–217, Montreal, Canada. AACE.

[Conlan et al., 2006] Conlan, O., Wade, V., Bruen, C., and Gargan, M. (2006). Multi-model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning. In De Bra, P., Brusilovsky, P., and Conejo, R., editors, *Adaptive Hypermedia and Adaptive Web-Based Systems*, volume 2347 of *Lecture Notes in Computer Science*, pages 100–111. Springer Berlin / Heidelberg.

[Conlan and Wade, 2004] Conlan, O. and Wade, V. P. (2004). Evaluation of APeLS An Adaptive eLearning Service Based on the Multi-model, Metadata-Driven Approach. In De Bra, P. and Nejdl, W., editors, *Adaptive Hypermedia and Adaptive Web-Based Systems*, volume 3137 of *Lecture Notes in Computer Science*, pages 504–518. Springer Berlin / Heidelberg.

[Cristea et al., 2005] Cristea, A., Smits, D., and de Bra, P. (2005). Writing MOT, Reading AHA! - converting between an authoring and a delivery system for adaptive educational hypermedia. In *The Third International Workshop on Authoring of Adaptive and Adaptable Educational Hypermedia*, Amsterdam, The Netherlands. AIED.

[Cristea and de Mooij, 2003] Cristea, A. I. and de Mooij, A. (2003). LAOS: Layered WWW AHS Authoring Model and their corresponding Algebraic Operators. In *WWW'03*, Budapest Hungary. ACM.

[Cryer and Gunn, 2008] Cryer, H. and Gunn, D. (2008). Exploring tactile graphics - Which strategies work? Technical report, RNIB Centre for Accessible Information.

[Dagger, 2006] Dagger, D. (2006). *Personalised eLearning Development Environments*. PhD thesis, Department of Computer Science, Trinity College Dublin.

[Dagger et al., 2003] Dagger, D., Conlan, O., and Wade, V. (2003). An Architecture for Candidacy in Adaptive eLearning Systems to Facilitate the Reuse of Learning Resources. In Rossett, A., editor, *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2003*, pages 49–56, Phoenix, Arizona, USA. AACE.

[Dagger et al., 2005] Dagger, D., Wade, V., and Conlan, O. (2005). Personalisation for All: Making Adaptive Course Composition Easy. *Educational Technology & Society*, 8(3):9–25.

[DAISY, 2005] DAISY (2005). DAISY 3 Specification. `http://www.daisy.org/z3986/2005/Z3986-2005.html`.

[DAISY, 2010] DAISY (2010). Daisy Consortium Website. `http://www.daisy.org/daisy-standard`.

[DCMI, 2004] DCMI (2004). Dublin Core Metadata Initiative. `http://dublincore.org/`.

[DCMI, 2010a] DCMI (2010a). DCMI Type Vocabulary. `http://dublincore.org/documents/dcmi-type-vocabulary/`.

[DCMI, 2010b] DCMI (2010b). Dublin Core Metadata Element Set. `http://dublincore.org/documents/dces/`.

[de Moura et al., 2005] de Moura, S. L., Coutinho, F., Siqueira, S. W. M., Melo, R. N., and Nunes Jr., S. V. (2005). Integrating Repositories of Learning Objects Using Web-Services to Implement Mediators and Wrappers. In *Proceedings of the International Conference on Next Generation Web Services Practices*, pages 82–87, Washington, DC, USA. IEEE Computer Society.

[DeBra et al., 2002] DeBra, P., Aerts, A. T. M., Smits, D., and Stash, N. (2002). AHA! meets AHAM. In *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, AH '02, pages 388–391, London, UK. Springer-Verlag.

[DeBra et al., 1999] DeBra, P., Houben, G.-J., and Wu, H. (1999). AHAM: a Dexter-based Reference Model for Adaptive Hypermedia. In *Proceedings of the tenth ACM Conference on Hypertext and hypermedia : returning to our diverse roots: returning to our diverse roots*, HYPERTEXT '99, pages 147–156, New York, NY, USA. ACM.

[Dodds, 1988] Dodds, A. G. (1988). Tactile maps and the blind user: perceptual, cognitive and behavioural factors. In *Proceedings of the Second International Symposium on Maps and Graphics for Visually Handicapped People*.

[E-Learn-VIP, 2010] E-Learn-VIP (2010). E-Learning for Visually Impaired Persons Consortium Website. `http://www.e-learn-vip.org/`.

[Edman, 1992] Edman, P. K. (1992). *Tactile Graphics*. American Foundation for the Blind, New York.

[ELF, 2010] ELF (2010). E-Learning Framework Website. `elframework.org`.

[ETSI, 2010] ETSI (2010). User profile preferences and information. `http://portal.etsi.org/stfs/STF_HomePages/STF342/es_202746v010101p.doc`.

[eXist, 2010] eXist (2010). Open Source Native XML Database. `http://exist.sourceforge.net/`.

[Felder and Silverman, 1988] Felder, R. M. and Silverman, L. K. (1988). Learning And Teaching Styles In Engineering Education. *Engineering Education*, 78(7):674–681.

[Fisseler and Bühler, 2007] Fisseler, B. and Bühler, C. (2007). Accessible E-Learning and Educational Technology Extending Learning Opportunities for People with Disabilities. In *International Conference on Interactive computer aided learning: EPortfolio and Quality in e-Learning*.

[Fitzpatrick and McMullen, 2008] Fitzpatrick, D. and McMullen, D. (2008). Distance Learning of Graphically Intensive Material for Visually Impaired Students. In *International Conference on Computers Helping People with Special Needs*, volume 5105, pages 219–225. Springer Berlin / Heidelberg.

[Frankola, 2001] Frankola, K. (2001). Why Online Learners Drop Out. *Workforce*, 80(10).

[Froschl, 2005] Froschl, C. (2005). User Modeling and User Profiling in Adaptive E-learning Systems. Master's thesis, Graz University of Technology, Graz Austria.

[Gardner, 1996] Gardner, J. (1996). Tactile Graphics: an Overview and Resource Guide. *Information Technology*.

[Gardner and Bulatov, 1997] Gardner, J. and Bulatov, V. (1997). The Problem of Accessing Non-Textual Information On The Web. *Proceedings of ths 1997 Conference on the W3 Consortium*.

[Gardner and Bulatov, 2006] Gardner, J. A. and Bulatov, V. (2006). Scientific Diagrams Made Easy with IVEO. In *International Conference on Computers Helping People with Special Needs*, volume 4061, pages 1243 – 1250.

[Gardner et al., 2005] Gardner, J. A., Bulatov, V., and Stowell, H. (2005). The Viewplus IVEO Technology For Universally Usable Graphical Information. *Proceedings of the 2005 CSUN International Conference on Technology and People with Disabilities*.

[Gardner et al., 2008] Gardner, J. A., Gardner, C. K., Jones, B., and Jones, E. (2008). Making Arithmetic Accessible. In *Proceedings of the 11th international conference on Computers Helping People with Special Needs*, ICCHP '08, pages 900–906, Berlin, Heidelberg. Springer-Verlag.

[GEM, 2010] GEM (2010). GEM Consortium. *http://www.gemconsortium.org/*.

[Ghelman et al., 2006] Ghelman, R., Siqueira, S. W. M., and Melo, R. N. (2006). Providing Accessibility to Distributed and Heterogeneous Learning Objects. In *Euro American Conference on Telematics and Information Systems*, Santa Martha, Colombia.

[GlassFish, 2010] GlassFish (2010). Sun GlassFish Enterprise Server. *http://glassfish.java.net/*.

[Gomes et al., 2005] Gomes, G. R. R., Siqueira, S. W. M., Braz, M. H. L. B., and Melo, R. N. (2005). LORDiLIS: Integrating Learning Objects Repositories and Digital Libraries. In *Consolidation eLearning Experiences*, Sao Paolo. Getulio Vargas Foundation.

[GRAB, 2010] GRAB (2010). GRAB Project Website. *http://www.grab-eu.com/id15.htm*.

[GWMicro, 2010] GWMicro (2010). Window-Eyes. *http://www.gwmicro.com/Window-Eyes/*.

[H3DAPI, 2010] H3DAPI (2010). Open Source Haptics. *http://www.h3dapi.org/*.

[Halasz and Schwartz, 1994] Halasz, F. and Schwartz, M. (1994). The Dexter hypertext reference model. *Commun. ACM*, 37:30–39.

[Hart and Staveland, 1988] Hart, S. and Staveland, L. (1988). *Human Mental Workload*, chapter Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research, pages 139–183. North-Holland.

[Herring, 2007] Herring, N. (2007). Considerations for Embosser Produced Tactile Graphics. Technical report, Pentronics.

[Hoy et al., 2002] Hoy, M., Wood, D., Loy, M., Elliot, J., and Eckstein, R. (2002). *Java Swing*. O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2 edition.

[HTTP, 2010] HTTP (2010). Hypertext Transfer Protocol. *http://www.w3.org/Protocols/*.

[HumanWare, 2010] HumanWare (2010). HumanWare Website. *http://www.humanware.com/en-united_kingdom/products/blindness/braille_displays*.

[HyperBraille, 2010] HyperBraille (2010). HyperBraille Project Website. *http://www.hyperbraille.de/?lang=en*.

[IEEE, 1997] IEEE (1997). Public and Private Information for Learners. *http://www.cen-ltso.net/main.aspx?put=230*.

[IEEE, 2001] IEEE (2001). Learning Technology Systems Architecture. *http://ltsc.ieee.org/wg1/files/IEEE_1484_01_D09_LTSA.pdf*.

[IEEE, 2002] IEEE (2002). Learning Object Metadata. *http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf*.

[IEEE, 2007] IEEE (2007). Web Services Description Language. *http://www.w3.org/TR/wsdl20/*.

[IMS, 2001] IMS (2001). Learning Resource Metadata. *http://www.imsglobal.org/metadata/*.

[IMS, 2003a] IMS (2003a). Learning Design. *http://www.imsglobal.org/learningdesign/*.

[IMS, 2003b] IMS (2003b). Simple Sequencing. *http://www.imsglobal.org/simplesequencing/*.

[IMS, 2005a] IMS (2005a). Learner Information Package. *http://www.imsglobal.org/profiles/*.

[IMS, 2005b] IMS (2005b). Learner Information Package Specification. *http://www.imsglobal.org/accessibility/*.

[IMS, 2007] IMS (2007). Content Packaging Specification. *http://www.imsglobal.org/content/packaging/*.

[IMS, 2010] IMS (2010). Global Learning Consortium. *http://www.imsglobal.org/*.

[ISO, 2008] ISO (2008). Individualized adaptability and accessibility in e-learning, education and training. *http://www.iso.org/iso/pressrelease.htm?refid=Ref1217*.

[JAVA, 2010] JAVA (2010). JAVA Website. *http://www.oracle.com/technetwork/java/index.html*.

[JAWS, 2010] JAWS (2010). JAWS Website. *http://www.freedomscientific.com/products/fs/jaws-product-page.asp*.

[JAX-WS, 2006] JAX-WS (2006). JAX-WS API. *http://java.sun.com/developer/technicalArticles/J2SE/jax_ws_2/*.

[JSAPI, 2010] JSAPI (2010). JAVA Speech API. *http://java.sun.com/products/java-media/speech/*.

[Kazman et al., 1994] Kazman, R., Bass, L., Webb, M., and Abowd, G. (1994). Saam: a method for analyzing the properties of software architectures. In *Proceedings of the 16th international conference on Software engineering*, ICSE '94, pages 81–90, Los Alamitos, CA, USA. IEEE Computer Society Press.

[Kazman et al., 1998] Kazman, R., Klein, M., Barbacci, M., Longstaff, T., Lipson, H., and Carriere, J. (1998). The Architecture Tradeoff Analysis Method. *Engineering of Complex Computer Systems, IEEE International Conference on*, 0:0068.

[Kennedy et al., 1991] Kennedy, J., Gabias, P., and Nicholls, A. (1991). *The Psychology of Touch*, chapter Tactile Pictures. Lawrence Erlbaum Associates.

[Koper et al., 2004] Koper, R., Olivier, B., St, D., and Ab, B. B. (2004). Representing the Learning Design of Units of Learning. *Educational Technology & Society*, 7:97–111.

[Kraus et al., 2008] Kraus, M., Völkel, T., and Weber, G. (2008). An Off-Screen Model for Tactile Graphical User Interfaces. In Miesenberger, K., Klaus, J., Zagler, W., and

Karshmer, A., editors, *Computers Helping People with Special Needs*, volume 5105 of *Lecture Notes in Computer Science*, pages 865–872. Springer Berlin / Heidelberg.

[Landau et al., 2008] Landau, S., Bourquin, G., van Schaak, A., and Miele, J. (2008). Demonstration of a Universally Accessible Audio-Haptic Transit Map Built on a Digital Pen-Based Platform. *3rd International Haptic and Auditory Interaction Design Workshop*, 2:23–24.

[Landau and Gourgey, 2001] Landau, S. and Gourgey, K. (2001). Development Of A Tactile Tactile Tablet. *Information Technology and Disabilities*, VIII.

[Landau and Gourgey, 2004] Landau, S. and Gourgey, K. (2004). A New Approach to Interactive Audio/Tactile Computing: The Talking Tactile Tablet. *California State University Northridge, Center On Disabilities Technology And Persons With Disabilities Conference*.

[Landau and Wells, 2003] Landau, S. and Wells, L. (2003). Merging Tactile Sensory Input and Audio Data by Means of The Talking Tactile Tablet. *Eurohaptics 2003*.

[LiveScribe, 2010] LiveScribe (2010). LiveScribe Website. `http://www.livescribe.com/en-us/`.

[Loetzsch, 1994] Loetzsch, J. (1994). Computer-aided access to tactile graphics for the blind. In Zagler, W., Busby, G., and Wagner, R., editors, *Computers for Handicapped Persons*, volume 860 of *Lecture Notes in Computer Science*, pages 575–581. Springer Berlin / Heidelberg.

[Loetzsch and Roedig, 1996] Loetzsch, J. and Roedig, G. (1996). Interactive Tactile Media in Training Visually Handicapped People. In Burger, D., editor, *New Technologies in the Education of the Visually Handicapped*, volume 237, pages 155–160, Montrouge, France.

[LOLA, 2010] LOLA (2010). LOLA Website. `http://www.nottingham.ac.uk/nursing/lola/`.

[LSAL, 2010] LSAL (2010). Learning Systems Architecture Lab. `lsal.org`.

[Macromedia, 2010] Macromedia (2010). Macromedia Director. `http://www.adobe.com/products/director/`.

[McMullen, 2008] McMullen, D. (2008). Multimodal Access to Visually Intensive Material at a Distance. In *International Conference on Computers Helping People with Special Needs, Doctoral Consortium*.

[McMullen, 2010] McMullen, D. (2010). A New Approach for the Production and Delivery of Audio Tactile Diagrams. In *International Conference on Computers Helping People with Special Needs, Doctoral Consortium*.

[McMullen and Fitzpatrick, 2008] McMullen, D. and Fitzpatrick, D. (2008). Tactile Diagrams at a Distance: A Multimodal E-learning Prototype. In Carthy, J. M., Pitt, I., and Kirakowski, J., editors, *Irish Human-computer Interaction Conference*, pages 101–105. University Of Limerick.

[McMullen and Fitzpatrick, 2009] McMullen, D. and Fitzpatrick, D. (2009). Autonomous Access to Graphics for Visually Impaired Learners. In *EdTech*.

[McMullen and Fitzpatrick, 2010a] McMullen, D. and Fitzpatrick, D. (2010a). Combining Web Services and DAISY for the Production and Delivery of Audio Tactile Diagrams. In *ICCHP (1)*, pages 12–19.

[McMullen and Fitzpatrick, 2010b] McMullen, D. and Fitzpatrick, D. (2010b). Service Based Approach to the Construction and Delivery of Audio Tactile Diagrams. *California State University Northridge, Center On Disabilities Technology And Persons With Disabilities Conference*.

[Meister, 2002] Meister, J. (2002). Pillars of E-learning Success. Technical report, Corporate University Exchange, New York.

[Melia, 2009] Melia, M. (2009). *Constraint-based validation of e-learning courseware*. PhD thesis, School of Computing, Dublin City University.

[MERLOT, 2010] MERLOT (2010). Multimedia Educational Resource for Learning and Online Teaching. `http://www.merlot.org/merlot/index.htm`.

[MICOLE, 2010] MICOLE (2010). MICOLE Project Website. `http://micole.cs.uta.fi/`.

[Miele, 2007] Miele, J. (2007). The Live Scribe Smartpen: A Revolutionary New Platform for Audio/Tactile Graphics. In *International Cartographic Conference*.

[Miele and Landau, 2010] Miele, J. and Landau, S. (2010). Audio-Tactile Interactive Computing with the Livescribe Pulse Pen. *California State University Northridge, Center On Disabilities Technology And Persons With Disabilities Conference*.

[Miele et al., 2006] Miele, J., Landau, S., and Gilden, D. (2006). Talking TMAP: Automated generation of audio-tactile maps using Smith-Kettlewell's TMAP software. *Bri*, 24:93–100.

[Miele and Schaack, 2008] Miele, J. A. and Schaack, A. V. (2008). Audiotactile Graphics Using Mainstream Smartpen Technology: A New Approach with Enormous Potential. *California State University Northridge, Center On Disabilities Technology And Persons With Disabilities Conference*.

[Moodle, 2010] Moodle (2010). Moodle Website. `http://moodle.org/`.

[Moroney et al., 1992] Moroney, W. F., Biers, D. W., Eggemeier, F. T., and Mitchell, J. A. (1992). A comparison of two scoring procedures with the NASA task load index in a simulated flight task. In *Aerospace and Electronics Conference*, volume 2, pages 734–740.

[MySQL, 2010] MySQL (2010). MySQL Website. `http://www.mysql.com/`.

[NASA, 2010] NASA (2010). Task Load Index. `http://humansystems.arc.nasa.gov/groups/TLX/downloads/TLX.pdf`.

[NCTD, 2010] NCTD (2010). National Centre for Tactile Diagrams. `http://www.nctd.org.uk/`.

[NDLR, 2010] NDLR (2010). National Digital Learning Resources. `http://www.ndlr.ie/`.

[Nolan and Morris, 1971] Nolan, C. and Morris, J. (1971). Improvement of tactual symbols for blind children: Final report. Technical report, American Printing House for the Blind.

[Novint, 2010] Novint (2010). Novint Falcon. `http://home.novint.com/products/novint_falcon.php`.

[Nygren, 1991] Nygren, T. E. (1991). Psychometric properties of subjective workload measurement techniques: implications for their use in the assessment of perceived mental workload. *Hum. Factors*, 33:17–33.

[Parkes, 1988] Parkes, D. (1988). Nomad, an audio-tactile tool for the acquisition, use and management of spatially distributed information by visually impaired people. In Tatham, A. F. and Dodds, A. G., editors, *Proceedings of the 2nd Internation Symposium on Maps and Graphics for Visually Impaired People*, pages 24–29, Kings College, University of London.

[Parkes, 1994] Parkes, D. (1994). Audio tactile systems for designing and learning complex environments as a vision impaired person: Static and dynamic spatial information access. In Steele, J. and Hedberg, J. G., editors, *Learning Environment Technology: Selected papers from LETA 94*, pages 219–223.

[Pennisi, 1992] Pennisi, E. (1992). Talking maps: technologies to give the visually impaired a sense of space. *Science News*.

[Pentronics, 2010] Pentronics (2010). PictureBraille Website. `http://www.pentronics.com.au/index_files/PictureBraille.htm`.

[PHP, 2010] PHP (2010). PHP: Hypertext Processor. `http://www.php.net/`.

[Pietrzak et al., 2007] Pietrzak, T., Martin, B., Pecci, I., Saarinen, R., Raisamo, R., and Järvi, J. (2007). The MICOLE Architecture: Multimodal Support for Inclusion of Visually Impaired Children. In *Proceedings of the 9th international conference on Multimodal interfaces*, ICMI '07, pages 193–200, New York, NY, USA. ACM.

[Polson and Richardson, 1988] Polson, M. C. and Richardson, J. J. (1988). *Foundations of intelligent tutoring systems*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA.

[PRCVI, 2010] PRCVI (2010). Provincial Resource Centre for the Visually Impaired. *http://www.prcvi.org/visualimpairment/file_download.html*.

[QTI, 2006] QTI, I. (2006). Question and Test Interoperability. *http://www.imsglobal.org/question/*.

[Rehak et al., 2005] Rehak, D. R., Dodds, P., and Lannom, L. (2005). A Model and Infrastructure for Federated Learning Content Repositories. In *In WWW 2005: International World Wide Web Conference*, pages 10–14.

[RELOAD, 2010] RELOAD (2010). RELOAD Project Website. *http://www.reload.ac.uk/*.

[Rosenblum et al., 2004] Rosenblum, L. P., Hasty, L., and Landau, S. (2004). TTT Authoring Tool Design Guide. Technical report, Touch Graphics, New York.

[SAPI, 2010] SAPI (2010). Microsoft Speech API. *http://msdn.microsoft.com/en-us/library/ms723627(VS.85).aspx*.

[Schmidt and Weber, 2009] Schmidt, M. and Weber, G. (2009). Multitouch Haptic Interaction. In Stephanidis, C., editor, *Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments*, volume 5615 of *Lecture Notes in Computer Science*, pages 574–582. Springer Berlin / Heidelberg.

[SEI, 2005] SEI (2005). Quality Attributes and Service-Oriented Architectures. *http://www.sei.cmu.edu/reports/05tn014.pdf*.

[Sensable, 2010a] Sensable (2010a). Phantom Desktop. *http://www.sensable.com/haptic-phantom-desktop.htm*.

[Sensable, 2010b] Sensable (2010b). Phantom Omni. *http://www.sensable.com/haptic-phantom-omni.htm*.

[Sensable, 2010c] Sensable (2010c). Phantom Premium. *http://www.sensable.com/haptic-phantom-premium.htm*.

[Sheppard and Aldrich, 2001] Sheppard, L. and Aldrich, F. K. (2001). Tactile graphics in school education: perspectives from teachers. *British Journal of Visual Impairment*, 19:93–97.

[Shneiderman, 1998] Shneiderman, B. (1998). *Designing the user interface: strategies for effective human-computer interaction*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[Siqueira et al., 2008] Siqueira, S. W., Braz, M. H. L., and Melo, R. N. (2008). *Architecture Solutions for E-Learning Systems*, chapter Accessibility, Digital Libraries and Semantic Web Standards in an E-Learning Architecture, pages 137–153. IGI Global.

[Siqueira et al., 2003] Siqueira, S. W. M., Braz, M. H. L. B., and Melo, R. N. (2003). Web Technology for Education and Training. *Database and Expert Systems Applications, International Workshop on*, 0:337.

[Stevens and Edwards, 1996] Stevens, R. D. and Edwards, A. D. N. (1996). An approach to the evaluation of assistive technology. In *Proceedings of the second annual ACM conference on Assistive technologies*, Assets '96, pages 64–71, New York, NY, USA. ACM.

[TactileLibrary, 2010] TactileLibrary (2010). Tactile Library. `http://www.tactilelibrary.com/`.

[TechSmith, 2010] TechSmith (2010). Camtasia. `http://www.techsmith.com/camtasia/`.

[TouchGraphics, 2010a] TouchGraphics (2010a). T3 Website. `http://www.touchgraphics.com/catalog/product_info.php?cPath=1&products_id=33`.

[TouchGraphics, 2010b] TouchGraphics (2010b). Touch Graphics Website. `http://www.touchgraphics.com/`.

[Tullis and Albert, 2008] Tullis, T. and Albert, B. (2008). *Measuring the User Experience*. Morgan Kaufmann.

[Türker et al., 2006] Türker, A., Görgün, I., and Conlan, O. (2006). The Challenge of Content Creation to Facilitate Personalized E-Learning Experiences. *International Journal on E-Learning*, 5(1):11–17.

[Ubuntu, 2010] Ubuntu (2010). Ubuntu Linux. `http://www.ubuntu.com/`.

[Ungar et al., 1997] Ungar, S., Blades, M., and Spencer, C. (1997). The use of tactile maps to aid navigation by blind and visually impaired people in unfamiliar urban environments. *Proceedings of the 1997 Conference of the Royal Institute of Navigation*.

[Verbert and Duval, 2007] Verbert, K. and Duval, E. (2007). Evaluating the ALOCOM Approach for Scalable Content Repurposing. In *EC-TEL*, pages 364–377.

[ViewPlus, 2009a] ViewPlus (2009a). Tiger Embosser. `http://www.viewplus.com/products/braille-embossers/braille-printers/elite-premier-braille-printer/`.

[ViewPlus, 2009b] ViewPlus (2009b). ViewPlus IVEO Creator 2.0 and Creator Pro 2.0 Manual. Technical report, ViewPlus.

[ViewPlus, 2010a] ViewPlus (2010a). IVEO Website. `http://www.viewplus.com/products/touch-audio-learning/IVEO/`.

[ViewPlus, 2010b] ViewPlus (2010b). Tiger Emprint Spotdot. `http://www.viewplus.com/products/braille-embossers/ink-braille-printers/emprint/`.

[ViewPlus, 2010c] ViewPlus (2010c). ViewPlus Website. `http://www.viewplus.com/`.

[Völkel et al., 2008] Völkel, T., Weber, G., and Baumann, U. (2008). Tactile Graphics Revised: The Novel BrailleDis 9000 Pin-Matrix Device with Multitouch Input. In Miesenberger, K., Klaus, J., Zagler, W., and Karshmer, A., editors, *Computers Helping People with Special Needs*, volume 5105 of *Lecture Notes in Computer Science*, pages 835–842. Springer Berlin / Heidelberg.

[W3C, 2004a] W3C (2004a). Web Services Glossary. *http://www.w3.org/TR/ws-gloss/*.

[W3C, 2004b] W3C (2004b). XML Schema. *http://www.w3.org/XML/Schema*.

[W3C, 2007a] W3C (2007a). Simple Object Access Protocol. *http://www.w3.org/TR/soap/*.

[W3C, 2007b] W3C (2007b). Web Services Description Language. *http://www.w3.org/TR/wsdl20/*.

[W3C, 2007c] W3C (2007c). XQuery. *http://www.w3.org/TR/xquery/*.

[W3C, 2008] W3C (2008). Extensible Markup Language. *http://www.w3.org/XML/*.

[W3C, 2009] W3C (2009). Extensible MultiModal Annotation markup language. *http://www.w3.org/TR/emma/*.

[W3C, 2010] W3C (2010). Scalable Vector Graphics. *http://www.w3.org/TR/SVG/*.

[w3schools, 2010] w3schools (2010). MIME Reference. *http://www.w3schools.com/media/media_mimeref.asp*.

[Wade and Doherty, 2000] Wade, V. P. and Doherty, P. (2000). A Meta-Data Driven Approach to Searching for Educational Resources in a Global Context. In Davies, G. and Owen, C. B., editors, *WebNet 2000 World Conference on the WWW and Internet*, pages 136–145. AACE.

[WAI, 2008] WAI (2008). Web Content Accessilibity Guidelines. *http://www.w3.org/TR/WCAG20/*.

[WAI, 2010] WAI (2010). Web Accessibility Initiative. *http://www.w3.org/WAI/*.

[Warmer and Kleppe, 2003] Warmer, J. and Kleppe, A. (2003). *The Object Constraint Language*. Addison-Wesley.

[Web3D, 2009] Web3D (2009). X3D. `http://www.web3d.org/x3d/`.

[Weber, 2010] Weber, G. (2010). ICCHP Keynote: Designing Haptic Interaction for a Collaborative World. In *ICCHP (2)*, pages 431–438.

[Wiley, 2000] Wiley, D. A. (2000). *Learning object design and sequencing theory*. PhD thesis, Brigham Young University.

[Wood et al., 2003] Wood, J., Magennis, M., Francisca, E., Arias, C., Gutierrez, T., and Bergamasco, M. (2003). The Design and Evaluation of a Computer Game for the Blind in the GRAB Haptic Audio Virtual Environment. *EuroHaptics*.

[Zeng and Weber, 2010] Zeng, L. and Weber, G. (2010). Audio-Haptic Browser for a Geographical Information System. In Miesenberger, K., Klaus, J., Zagler, W., and Karshmer, A., editors, *Computers Helping People with Special Needs*, volume 6180 of *Lecture Notes in Computer Science*, pages 466–473. Springer Berlin / Heidelberg.

Appendix A

XML Schemas

A.1 Visual Object Model XML Schema

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xs:schema xmlns:dc="http://purl.org/dc/elements/1.1/"
    attributeFormDefault="unqualified" elementFormDefault="unqualified"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import schemaLocation="http://dublincore.org/schemas/xmls/qdc
      /2008/02/11/dc.xsd" namespace="http://purl.org/dc/elements/1.1/" />
  <!-- definition of simple elements -->
  <xs:element name="content" type="xs:string" />
  <!-- definition of complex elements -->
  <xs:element name="metadata">
    <xs:complexType>
      <xs:all>
        <xs:element minOccurs="1" ref="dc:title" />
        <xs:element minOccurs="1" ref="dc:creator" />
        <xs:element minOccurs="1" ref="dc:subject" />
        <xs:element minOccurs="1" ref="dc:description" />
        <xs:element minOccurs="0" ref="dc:date" />
        <xs:element minOccurs="1" ref="dc:identifier" />
        <xs:element minOccurs="0" ref="dc:language" />
        <xs:element minOccurs="0" ref="dc:publisher" />
        <xs:element minOccurs="1" ref="dc:format" />
        <xs:element minOccurs="0" ref="dc:contributor" />
```

```
        <xs:element minOccurs="0" ref="dc:coverage" />
        <xs:element minOccurs="0" ref="dc:rights" />
        <xs:element minOccurs="0" ref="dc:relation" />
        <xs:element minOccurs="0" ref="dc:source" />
        <xs:element minOccurs="0" ref="dc:type" />
      </xs:all>
    </xs:complexType>
  </xs:element>
  <xs:element name="vo">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" ref="metadata" />
        <xs:element minOccurs="1" maxOccurs="1" ref="content" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Listing A.1: Visual Object Model XSD

A.2 Content Object Model XML Schema

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xs:schema xmlns:dc="http://purl.org/dc/elements/1.1/"
    attributeFormDefault="unqualified" elementFormDefault="unqualified"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import schemaLocation="http://dublincore.org/schemas/xmls/qdc
      /2008/02/11/dc.xsd" namespace="http://purl.org/dc/elements/1.1/" />
  <!-- definition of simple elements -->
  <xs:element name="title" type="xs:string" />
  <xs:element name="content" type="xs:string" />
  <xs:attribute name="mime-type" type="xs:string" />
  <!-- definition of complex elements -->
  <xs:element name="metadata">
    <xs:complexType>
      <xs:all>
```

```
        <xs:element minOccurs="1" ref="dc:title" />
        <xs:element minOccurs="1" ref="dc:creator" />
        <xs:element minOccurs="1" ref="dc:subject" />
        <xs:element minOccurs="1" ref="dc:description" />
        <xs:element minOccurs="0" ref="dc:date" />
        <xs:element minOccurs="1" ref="dc:identifier" />
        <xs:element minOccurs="1" ref="dc:language" />
        <xs:element minOccurs="0" ref="dc:publisher" />
        <xs:element minOccurs="0" ref="dc:format" />
        <xs:element minOccurs="0" ref="dc:contributor" />
        <xs:element minOccurs="0" ref="dc:coverage" />
        <xs:element minOccurs="0" ref="dc:rights" />
        <xs:element minOccurs="0" ref="dc:relation" />
        <xs:element minOccurs="0" ref="dc:source" />
        <xs:element minOccurs="0" ref="dc:type" />
      </xs:all>
    </xs:complexType>
  </xs:element>
  <xs:element name="fragment">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" ref="content" />
      </xs:sequence>
      <xs:attribute ref="mime-type" use="required" />
    </xs:complexType>
  </xs:element>
  <xs:element name="co">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" ref="metadata" />
        <xs:element minOccurs="1" maxOccurs="1" ref="title" />
        <xs:element minOccurs="0" maxOccurs="unbounded" ref="fragment" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Listing A.2: Content Object Model XSD

A.3 Assembly File Model XML Schema

```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<xs:schema xmlns:dc="http://purl.org/dc/elements/1.1/"
    attributeFormDefault="unqualified" elementFormDefault="unqualified"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import schemaLocation="http://dublincore.org/schemas/xmls/qdc
      /2008/02/11/dc.xsd" namespace="http://purl.org/dc/elements/1.1/" />
  <!-- definition of simple elements -->
  <!-- definition of attributes -->
  <xs:attribute name="wsdl" type="xs:string" />
  <xs:attribute name="namespace" type="xs:string" />
  <xs:attribute name="name" type="xs:string" />
  <xs:attribute name="id" type="xs:string" />
  <xs:attribute name="type" type="componentType" />
  <xs:attribute name="voElementId" type="xs:string" />
  <xs:attribute name="coId" type="xs:string" />
  <xs:simpleType name="componentType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="vo"/>
        <xs:enumeration value="co"/>
    </xs:restriction>
  </xs:simpleType>
  <!-- definition of complex elements -->
  <xs:element name="metadata">
    <xs:complexType>
      <xs:all>
        <xs:element minOccurs="1" ref="dc:title" />
        <xs:element minOccurs="1" ref="dc:creator" />
        <xs:element minOccurs="1" ref="dc:subject" />
        <xs:element minOccurs="1" ref="dc:description" />
        <xs:element minOccurs="0" ref="dc:date" />
        <xs:element minOccurs="1" ref="dc:identifier" />
```

```
      <xs:element minOccurs="1" ref="dc:language" />
      <xs:element minOccurs="0" ref="dc:publisher" />
      <xs:element minOccurs="0" ref="dc:format" />
      <xs:element minOccurs="0" ref="dc:contributor" />
      <xs:element minOccurs="0" ref="dc:coverage" />
      <xs:element minOccurs="0" ref="dc:rights" />
      <xs:element minOccurs="0" ref="dc:relation" />
      <xs:element minOccurs="0" ref="dc:source" />
      <xs:element minOccurs="0" ref="dc:type" />
    </xs:all>
  </xs:complexType>
</xs:element>
<xs:element name="service">
<xs:complexType>
      <xs:attribute ref="wsdl" use="required" />
      <xs:attribute ref="namespace" use="required" />
      <xs:attribute ref="name" use="required" />
</xs:complexType>
</xs:element>
<xs:element name="component">
  <xs:complexType>
  <xs:sequence>
     <xs:element minOccurs="1" maxOccurs="1" ref="service" />
   </xs:sequence>
    <xs:attribute ref="id" use="required" />
    <xs:attribute ref="type" use="required" />
  </xs:complexType>
</xs:element>
<xs:element name="components">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" ref="component" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="map">
  <xs:complexType>
```

```
      <xs:attribute ref="voElementId" use="required" />
      <xs:attribute ref="coId" use="required" />
    </xs:complexType>
  </xs:element>
  <xs:element name="mapping">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="map" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="coref">
    <xs:complexType>
      <xs:attribute ref="id" use="required" />
    </xs:complexType>
  </xs:element>
  <xs:element name="sequence">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="coref" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="assembly">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="metadata" minOccurs="1" maxOccurs="1"/>
        <xs:element ref="components" minOccurs="1" maxOccurs="1"/>
        <xs:element ref="mapping" minOccurs="1" maxOccurs="1"/>
        <xs:element ref="sequence" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Listing A.3: Assembly File Model XSD

Appendix B

Evaluation Documentation

B.1 Task 1 Tutorial

Procedure

This tutorial will provide instructions on how to assemble a Multimodal Graphic using new content.

Step 1:

The first step in creating a multimodal graphic is to create a new Visual Object. A Visual Object is the image that the multimodal graphic is to be based on. Visual Object functionality is located in the ***Visual Object Menu***. If you select ***"New Visual Object"*** from the menu a new window will appear. Here you can create your Visual Object based on an image file. You can assume the file was either created by you or for you by a tactile image designer. Click on the ***"Import"*** button. The file you want is called ***"Permanent Teeth.svg"*** and can be found in the **Images** folder on the computers Desktop. When you select it you will see a preview of the image. Add the following metadata:

Title: Permanent Teeth

Subject: Permanent Teeth

Description: Tactile image of permanent teeth

Creator: Biology Teacher

Click on **"Save"**.

Step 2:

The second step is to create a number of Content Objects and link them to regions of the graphic. A content object contains the information that will be played to the learner when they press on the specific regions of the graphic. Content Object functionality is located in the ***Content Object Menu***. Click on ***"New Content Object"***. The tool is now in Content Object creation mode. Click on the region of the graphic that the content object should be linked to. As an example, click on the region pointed to by the label **"Central Incisor"**. A new window will appear. Enter the following information:

Title: Central Incisor

Fragment 1: The central incisor grows in between 7 and 8 years of age.

Fragment 2: It is located near the midline of the face.

Subject: Permanent Teeth

Description: Content for the central incisor

Creator: Biology Teacher

Click on **"Save"**

Repeat the process by clicking on the region pointed to by the label **"Lateral Incisor"** and enter the following information:

Title: Lateral Incisor

Fragment 1: The lateral incisor grows in between 8 and 9 years of age.

Fragment 2: It has been adapted for cutting.

Subject: Permanent Teeth

Description: Content for the lateral incisor

Creator: Biology Teacher

Step 3:

The third step involves providing metadata for the multimodal graphic as a whole. Metadata can be entered using the **Edit Menu**. Select ***"Metadata"***. A new window will appear. Enter the following information:

Title: Permanent Teeth

Subject: Permanent Teeth

Description: Multimodal graphic for permanent teeth

Creator: Biology Teacher

Click on **"Save"**

Step 4:

The fourth step involving saving the multimodal graphic to a repository. The facility is located in the **File Menu**. Click on ***"Save"***. A new window opens. Click on ***"Ok"***.

Task 1 Tutorial Image

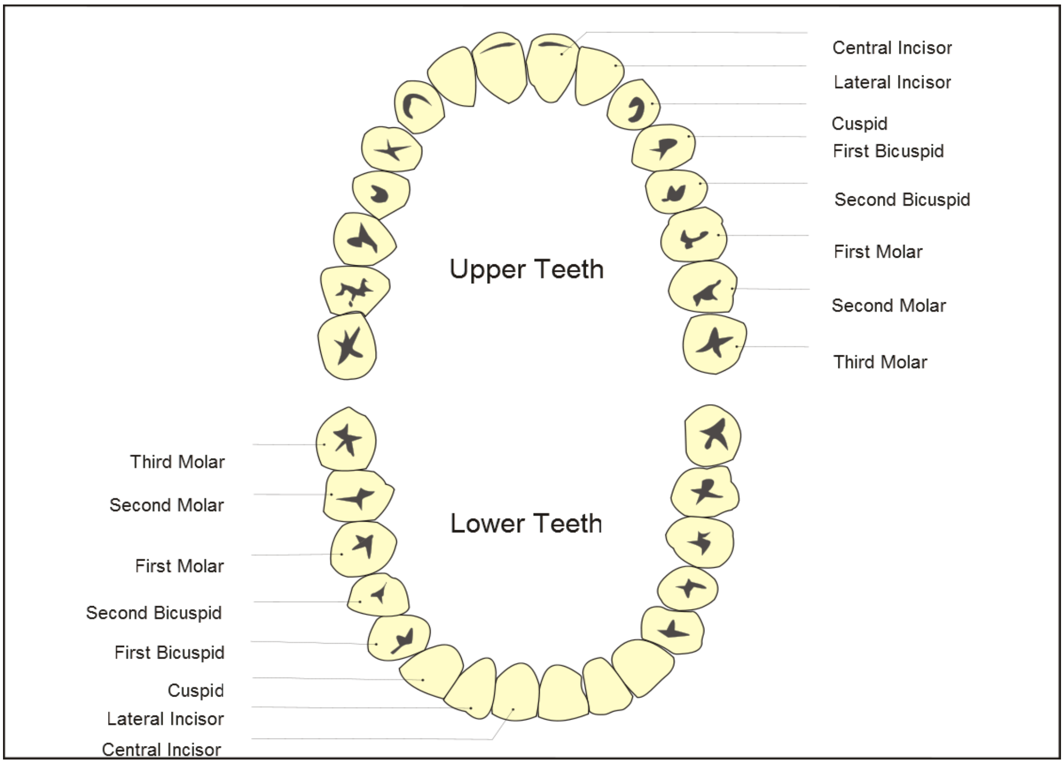


Figure B.1: Permanent Teeth

B.2 Task 2 Tutorial

Procedure

This tutorial will provide instructions on how to assemble a multimodal graphic using existing content. There are 4 steps.

Step 1:

The first step in creating a multimodal graphic is to search for a Visual Object. A Visual Object is the image that the multimodal graphic is to be based on. Visual Object functionality is located in the **Visual Object Menu**. If you select ***"Search for Visual Object"*** from the menu a new window will appear. Here you can use a keyword search to locate a Visual Object. Enter some keywords in the textbox labelled "Search Text". For example enter the words **"Nerve Cell"**. Click on the **"Search"** button. A list of results will appear at the bottom of the window. Click on one to preview the image on the right hand side. When you have found the Visual Object you want, click on, ***"Select"***. If the keyword search fails to work you can use the ***"List All"*** button.

Step 2:

The second step is to search for a number of Content Objects and link them to regions of the graphic. A content object contains the information that will be played to the learner when they press on the specific regions of the graphic. Content Object functionality is located in the **Content Object Menu**. Click on ***"Search for Content Object"***. The tool is now in Content Object search mode. Click on the region of the graphic that the content object should be linked to. As an example, click on the region pointed to by the label **"Myelin Sheath"**. A new window will appear similar to the Visual Object search window described above. Here you can use a keyword search to locate a Content Object. Enter some keywords in the textbox labelled "Search Text". To find suitable content for the region you clicked enter the words **"Myelin Sheath"**. Click on the ***"Search"*** button. A list of results will appear at the bottom of the window. Click on one to preview the content on the right hand side. When you have found the Content Object you want, click on, ***"Select"***. If the keyword search fails to work you can use the ***"List All"*** button. The Content Object has now been linked to the selected region. Repeat the process for the region pointed to by the label, **"Axon"**.

Step 3:

The third step involves providing metadata for the multimodal graphic as a whole. Metadata can be entered using the **Edit Menu**. Select ***"Metadata"***. A new window will appear. Enter the following information:

Title: Nerve Cell

Subject: Nerve Cell

Description: Multimodal graphic for nerve cell

Creator: Biology Teacher

Click on **"Save"**

Step 4:

The fourth step involving saving the multimodal graphic to a repository. The facility is located in the **File Menu**. Click on ***"Save"***. A new window opens. Click on ***"Ok"***.

Task 2 Tutorial Image

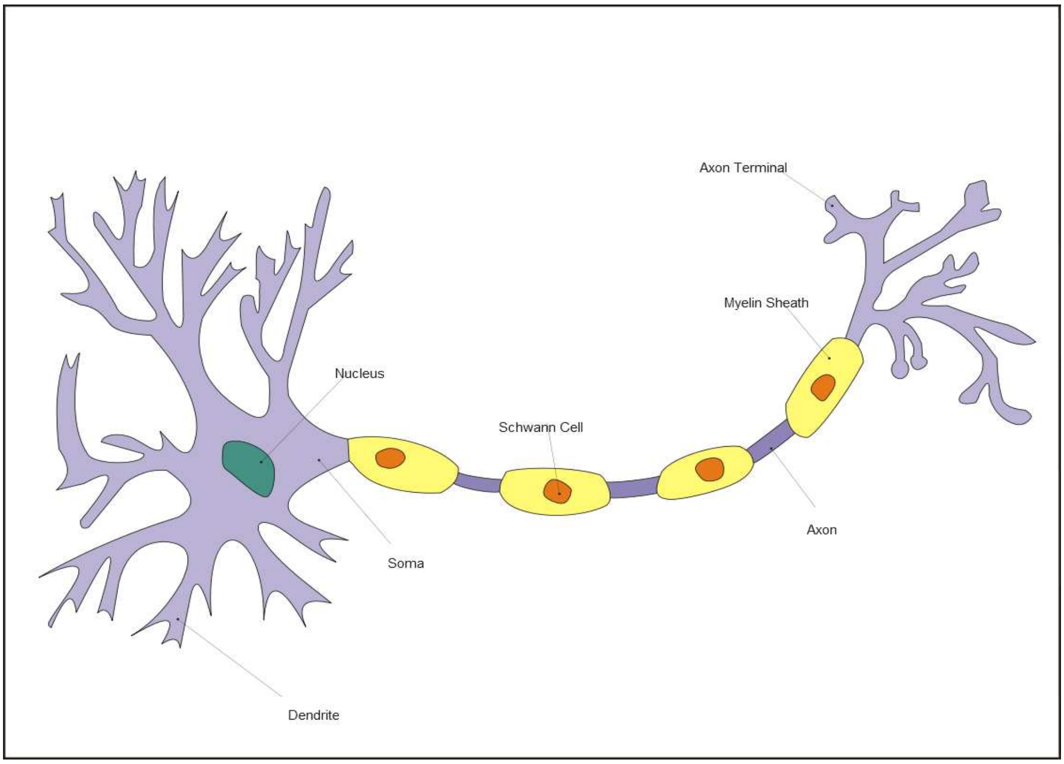


Figure B.2: Nerve Cell

B.3 Task 3 Tutorial

Procedure

Moodle is an E-Learning environment used in various institutions. Learning content is arranged in courses that the learners can explore. This tutorial will provide instructions on how to integrate a multimodal graphic into a Moodle course.

Step 1:

The first step in integrating a multimodal graphic into a Moodle course is to locate the activity menu. This is a dropdown menu containing the instruction **"Add an activity"**. Click on the menu and select the item that says **"MGADF"**.

Step 2:

A new window will open that allows you to search for a multimodal graphic. In the centre of the screen you will see a textbox with the label **"Keywords"**. Enter the words "Permanent Teeth" into the textbox and click on the **"Search"** button.

Step 3:

A list of results will appear at the bottom of the screen. If you have located a graphic you would like to use, click on the **"Select"** button at the end of relevant row.

Step 4:

The Name and Identifier textboxes at the top of the page should now contain entries that match the details of the graphic you selected in the results. If so, click on **"Save and return to course"**.

Step 5:

You should now see a link in the Moodle course with the same title as the graphic you selected.

Task 3 Tutorial Image

The Permanent Teeth image seen in figure B.1 was used in the tutorial for Task 3.

B.4 Task 1

Procedure

Step 1:

Create a new Visual Object using the ***Visual Object Menu***. The file for the task is called **"The Human Digestive System.svg"** and can be found in the **Images** folder on the computers **Desktop**. The following metadata can be used:

Title: The Human Digestive System

Subject: The Human Digestive System

Description: Tactile image of the human digestive system

Creator: Biology Teacher

Step 2:

Create and link a Content Object to three regions of the graphic using the ***Content Object Menu***. The Content Objects to be created are, **"Liver"**, **"Stomach"** and **"Large Intestine"**. The fragments and metadata for each Content Object can be found below.

Title: Liver

Fragment 1: The liver is the largest internal organ in the human body.

Fragment 2: The liver produces bile which helps aid in digestion.

Subject: The Human Digestive System

Description: Content for the liver

Creator: Biology Teacher

Title: Stomach

Fragment 1: The stomach is a J-shaped hollow organ located in the midsection of the body.

Fragment 2: The stomach contains acid which helps to break down large pieces of food.

Subject: The Human Digestive System

Description: Content for the stomach

Creator: Biology Teacher

Title: Large Intestine

Fragment 1: The large intestine completes the final stage of the digestion cycle.

Fragment 2: The large intestine absorbs water from indigestible food matter.

Subject: The Human Digestive System

Description: Content for the large intestine

Creator: Biology Teacher

Step 3:

Add metadata for the entire multimodal graphic using the Edit Menu. The metadata to use is:

Title: The Human Digestive System

Subject: The Human Digestive System

Description: Multimodal Graphic of the human digestive system

Creator: Biology Teacher

Step 4:

Save the multimodal graphic using the *File Menu*.

Inform the evaluator when you have completed the task.

Task 1 Image

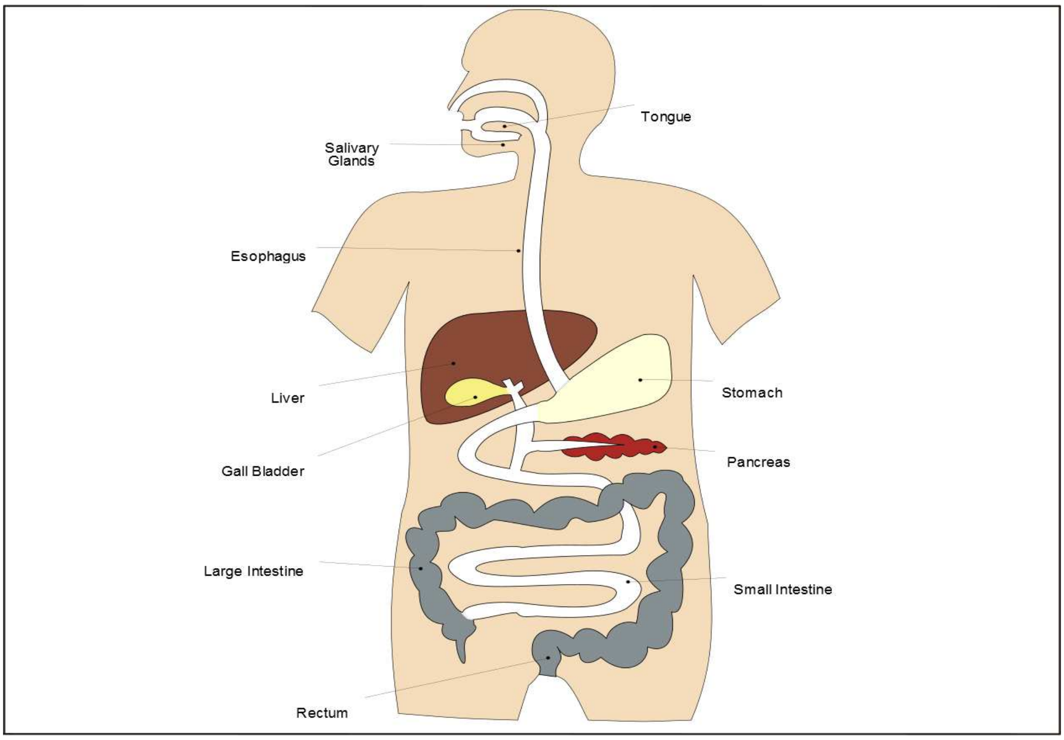


Figure B.3: The Human Digestive System

B.5 Task 2

Procedure

Step 1:

Search for a Visual Object using the ***Visual Object Menu***. The Visual Object you require is titled **"The Human Eye"**.

Step 2:

Search for and link a Content Object to three regions of the graphic using the ***Content Object Menu***. The Content Objects to be searched for are, **"Cornea"**, **"Lens"** and **"Vitreous Humour"**.

Step 3:

Add metadata for the entire multimodal graphic using the **Edit Menu**. The metadata to use is:

Title: The Human Eye

Subject: The Human Eye

Description: Multimodal Graphic of the human eye

Creator: Biology Teacher

Step 4:

Save the multimodal graphic using the **File Menu**.

Inform the evaluator when you have completed the task.

Task 2 Image

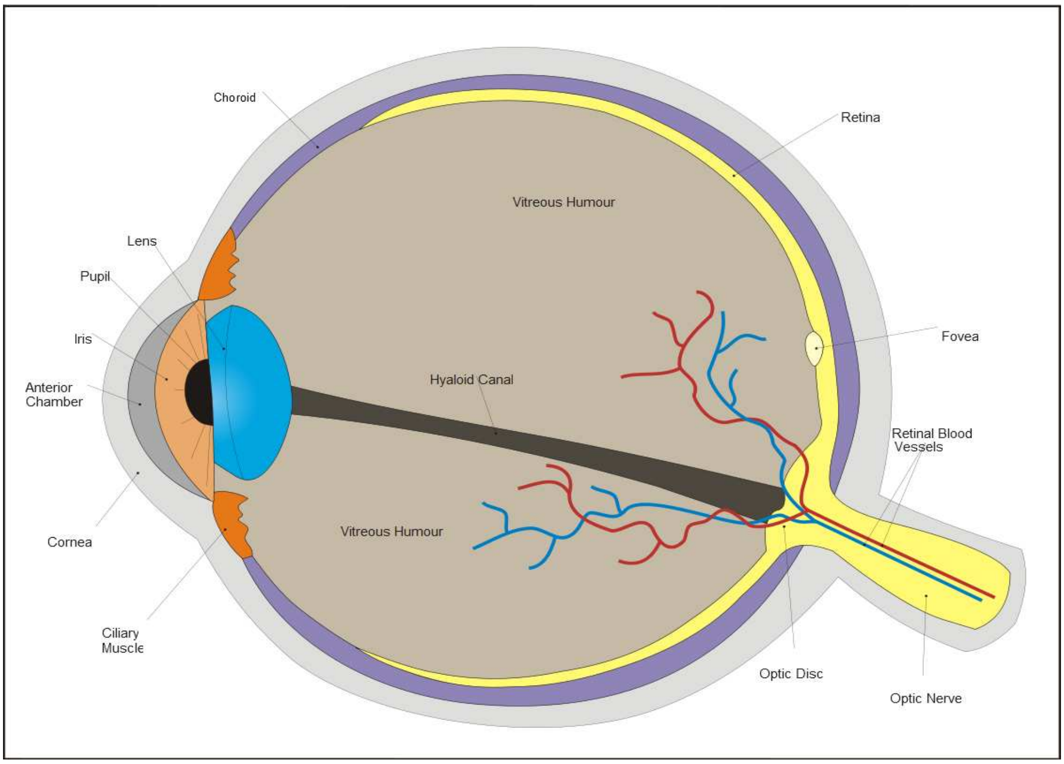
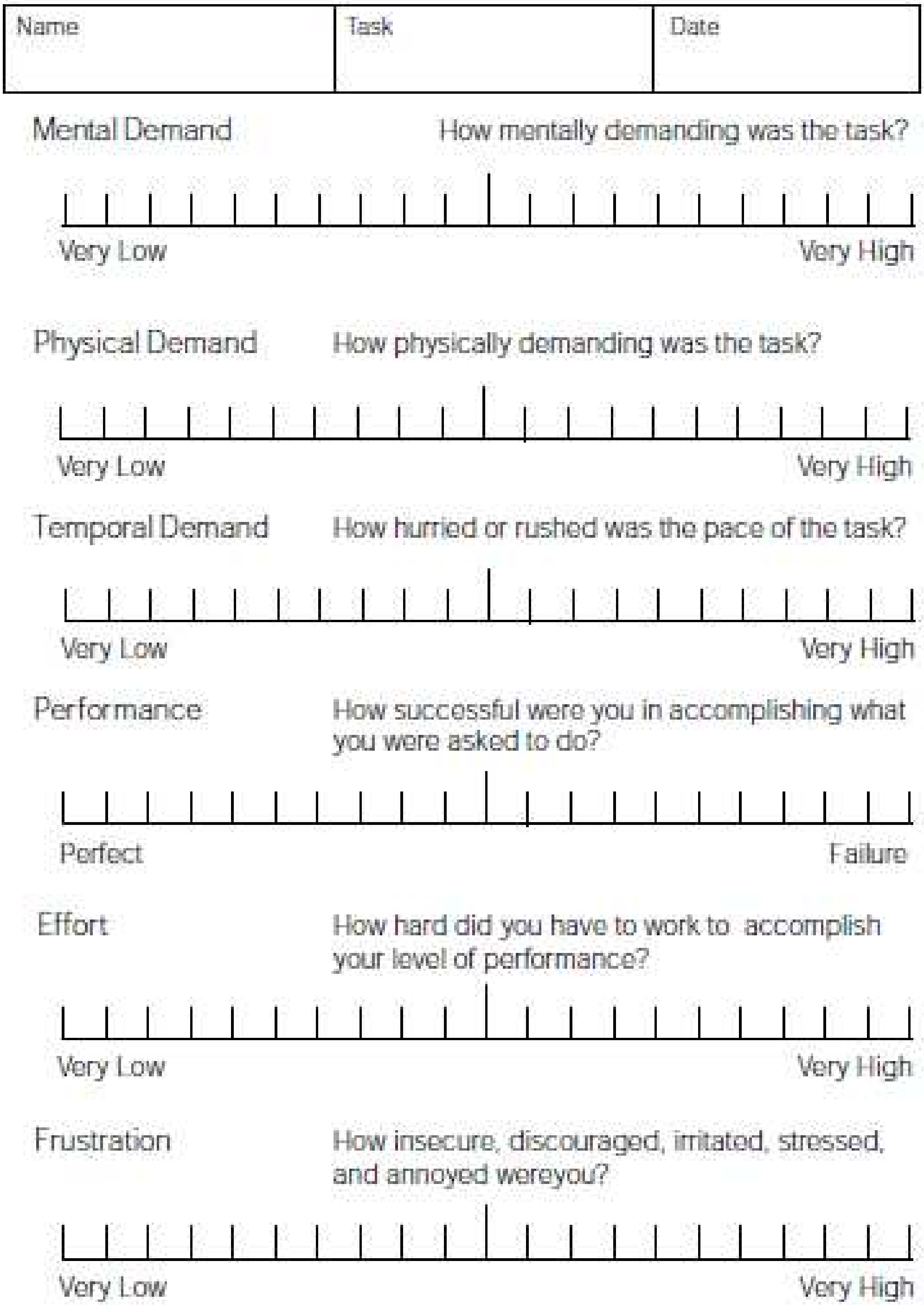


Figure B.4: The Human Eye

B.6 NASA TLX



Name	Task	Date

Mental Demand — How mentally demanding was the task?

Very Low — Very High

Physical Demand — How physically demanding was the task?

Very Low — Very High

Temporal Demand — How hurried or rushed was the pace of the task?

Very Low — Very High

Performance — How successful were you in accomplishing what you were asked to do?

Perfect — Failure

Effort — How hard did you have to work to accomplish your level of performance?

Very Low — Very High

Frustration — How insecure, discouraged, irritated, stressed, and annoyed wereyou?

Very Low — Very High

Figure B.5: NASA TLX

B.7 NASA TLX Rating Scale Definitions

MENTAL DEMAND

How much mental and perceptual activity was required (e.g. thinking, deciding, calculating, remembering, looking, searching etc)? Was the task easy or demanding, simple or complex, exacting or forgiving

PHYSICAL DEMAND

How much physical activity was required (e.g. pushing, pulling, turning, controlling, activating etc)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?

TEMPORAL DEMAND

How much time pressure did you feel due to the rate or pace at which the tasks or task element occurred? Was the pace slow and leisurely or rapid and frantic?

PERFORMANCE

How successful do you think you were in accomplishing the goals of the tasks set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?

EFFORT

How hard did you have to work (mentally and physically) to accomplish your level of performance?

FRUSTRATION

How insecure, discourages, irritated, stressed and annoyed versus secure, gratified, content relaxed and complacent did you feel during the task?

B.8 Questionnaire

This questionnaire consists of 16 questions. Some questions contain room for you to add additional comments if you wish. You will also get the opportunity to provide any feedback you desire on the evaluation at the end of the questionnaire. There are no right or wrong answers just you're honest opinion.

1. Participant Identifier
2. What is your gender? Male / Female
3. What is your age?
4. How would you describe your level of expertise with mainstream E-Learning? Very inexperienced / Inexperienced / Neutral / Experienced / Very experienced
5. How would you describe your level of expertise with multimodal graphic assembly? Very inexperienced / Inexperienced / Neutral / Experienced / Very experienced
6. Have you ever been required to teach a graphical concept to a Blind learner? Yes / No
7. If you answered Yes to the previous question, how did you illustrate the graphical concept to the Blind learners?
8. How difficult did you expect the process of multimodal graphic creation to be? Very difficult / Difficult / Neutral / Easy / Very easy
9. How difficult was it to create a multimodal graphic with the assembly tool? Very difficult / Difficult / Neutral / Easy / Very easy
10. The ability to search for and reuse existing content was beneficial. Strongly disagree / Disagree / Neutral / Agree / Strongly agree
11. I would prefer to create my own content than to search for suitable existing content. Strongly disagree / Disagree / Neutral / Agree / Strongly agree
12. I believe the metadata is expressive enough to facilitate searching. Yes / No

13. It was difficult to integrate a multimodal graphic into Moodle using the activity plugin. Strongly disagree / Disagree / Neutral / Agree / Strongly agree

14. I would use the system if I was required to provide graphics to Blind learners. Strongly disagree / Disagree / Neutral / Agree / Strongly agree

15. What are the weaknesses, (if any), that you feel the approach suffers from?

16. Please provide any additional comments you deem relevant.