UNIVERSITY OF CAPE TOWN
IYUNIVESITHI YASEKAPA • UNIVERSITEIT VAN KAAPSTAD

# An End-to-End Solution for Complex Open Educational Resources

*By:*
Morwan I. Mohamed Nour

*Supervisor:*
A/Prof. Hussein Suleman

digital libraries laboratory @uct.cs

Thesis presented for the Degree of Master of Science
In the Department of *Computer Science*
University of Cape Town

November 21, 2012

## Plagiarism Declaration:

1. I know that plagiarism is a serious form of academic dishonesty.
2. I have read the document about avoiding plagiarism, am familiar with its contents and have avoided all forms of plagiarism mentioned there.
3. Where I have used the words of others, I have indicated this by the use of quotation marks.
4. I have referenced all quotations and other ideas borrowed from others.
5. I have not and shall not allow others to plagiarise my work.

Signature:

*To* Mariam El-Hajj *and* Ismail Nur
*...It's all about you*

## Acknowledgements

A page full of words would not suffice to express the gratitude in me. Even though, I hope that everyone mentioned in the next few lines already knows how thankful I am for their help and support in all the different ways throughout this project and dissertation.

I need to start by thanking my parents, Ismail and Mariam, for their unconditional love and care. Your support, motivation and encouragement is invaluable and can never be measured in words. I owe it all to both of you. I would also like to thank my brothers and sisters for always being there.

I must acknowledge Michelle Willmers and Michael Paskevicius for sharing their vast knowledge, expertise and experiences in Open Educational Resources. I am grateful for all the time they spared to answer my questions and advise me in this project. I also appreciate all the contributions I've received from the Centre of Educational Technology.

A solid thanks to Stefano Rivera and Kyle Williams for their expertise, technical support and friendship.

A special thanks to my colleagues Jorgina Paihama and Lebeko Poulo for taking the road with me and making this an unforgettable experience.

To Michael Jackson: I truly appreciate your incredible assistance with the development.

I am very thankful to everyone in the Digital Libraries Laboratory for the amazing discussions, priceless knowledge sharing and all the good times in Room 300.

A million thanks to my dear *Cape Town Family*: Ivandro Issufo, Erika Rodrigues, Mayra and Erika Pereira, Alana Rafael, Fabio Dagot, Marina Chichava and Ana Kamya for making life in this city nothing but AWESOME!

Appreciation goes to my "brother", Ivan Pinto, for always keeping me in check and being by my side no matter what!

To Tracey van Heerden: thank you so much for being such an amazing friend and going through the last stretch with me.

I would like to forward my thanks to everyone at the Department of Computer Science at UCT for maintaining an exceptional environment for research.

Above all, special thanks to my supervisor, Hussein Suleman, for his continuous support and advice throughout the years. I can't express enough gratitude for his patience, availability, expertise and understanding. He always believed in me and kept inspiring me to this day. This work would not exist without him.

Finally, to everyone that I have not mentioned above, I did not forget. Thank you!

*"He who receives an idea from me, receives instruction himself without lessening mine; as he who lights his taper at mine, receives light without darkening me."*

THOMAS JEFFERSON, 1813

*"At the heart of the movement towards Open Educational Resources is the simple and powerful idea that the world's knowledge is a public good and that technology in general and the Worldwide Web in particular provide an opportunity for everyone to share, use, and reuse it."*

MIKE SMITH and CATHY CASSERLY, 2006

*"The rise of the Open Educational Resources movement is one of the most exciting, and indeed critical, developments of our time."*

BRENDA GOURLEY, 2008

*"The traditional forms of scientific publishing and peer review do not live up to the demands of efficient communication and quality assurance in today's highly diverse and rapidly evolving world of science. They need to be advanced by interactive and transparent forms of review, publication and discussion that are open to the scientific community and to the public."*

ULRICH PÖSCHL, 2010

**An End-to-end Solution for Complex Open Educational Resources**

**Abstract:** Open access and open resources have gained much attention from the world in the last few years. The interest in sharing information freely by the use of the World Wide Web has grown rapidly in many different fields.

Now, information is available in many different data forms because of the continuous evolution in technology. The main objective of this thesis is to provide content creators and educators with a solution that simplifies the process of depositing into digital repositories.

We created a desktop tool named ORchiD, Open educational Resources Depositor, to achieve this goal. The tool encompasses educational metadata and content packaging standards to create packages while conforming to a deposit protocol to ingest resources to repositories. A test repository was installed and adapted to handle Open Educational Resources.

The solution proposed is centered on the front-end application which handles the complex objects on the user desktop. The desktop application allows the user to select and describe his/her resource(s) then creates the package and forwards it to the specified repository using the deposit protocol.

The solution is proved to be simple for users but also in need of further improvements specifically in association to the metadata standard presented to user.

**Keywords:**

OER, EPrints, desktop tool, deposit protocol, Open Educational Resources, metadata standard

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| API | Application Programming Interface |
| DC | Dublin Core |
| IMS | International Metadata Standards |
| LOM | Learning Object Metadata |
| OER | Open Educational Resources |
| SCORM | Sharable Content Object Reference Model |
| SPI | Simple Publishing Interface |
| SWORD | Simple Web-service Offering Repository Deposit |
| XML | Extensible Markup Language |

Chapter 1

# Introduction

Contents



Open access and open resources have garnered much attention from the world in the last decade. The interest in sharing information freely by the use of the World Wide Web has grown rapidly in many fields. Now information is available in many file formats and complex structures [Atkins *et al.* 2007].

In 2002 during the UNESCO "Forum on the Impact of Open Courseware for Higher Education in Developing Countries" the term Open Education Resources (OER) was first adopted. OER was defined as:

> "*...technology-enabled, open provision of educational resources for consultation, use and adaptation by a community of users for non-commercial purposes. They are typically made freely available over the Web or the Internet. Their principal use is by teachers and educational institutions support course development, but they can also be used directly by students. Open Educational Resources include learning objects such as lecture material, references and readings, simulations, experiments and demonstrations,as well as syllabi, curricula and teacher guides.*" [UNESCO 2002]

OER is recognized by many as having the potential to advance the delivery of education. The concept does so by increasing availability of resources, reducing cost of educational materials and encouraging collaborative creation of learning objects by teaching staff and students [Butcher 2011].

## 1.1 Problem Statement and Motivation

There is a continuous need for simpler solutions to assist in most aspects of OER. A briefing [Yuan *et al.* 2008] paper exploring the challenges faced at the time expressed a current and long term need for more advanced tools and services for educational repositories. Increasing metadata creation challenges were also expressed in this paper. Some repositories are usually provided to encourage sharing, reuse, and repurposing of teaching and learning materials for many reasons. However, deposit activity remains weak or is still a problem [McGill *et al.* 2008]. A possible reason could be that the amount of effort that is currently required to deposit OERs may be a contributing factor to this problem.

The Open e-Learning Content Observatory Services (OLCOS) project also expressed that the need to implement more advanced tools and services for educational repositories can be considered as a long-term inhibitor for these repositories. Services such as deposit, create, recommend and annotate are examples of advancements that could help in the growth and up-take of repositories by the community depending on how they are implemented [Baumgartner *et al.* 2007].

As OER grew in popularity several projects and organisations have attempted to simplify the traditional repository deposit mechanism, which normally is a Web application and a set of forms for user entry. The main problem with this process is that its time consuming and different repository systems have unique requirements for resource submission, i.e., more work for the content depositor or user. Furthermore, it becomes tedious for a user to submit the same resource into multiple repositories. Several systems and applications attempt to integrate repositories with the desktop to simplify or enhance deposit of resources into repository systems. Some of these projects were created for specific file types and others for specific systems. These projects are diverse and some are directed at certain communities or organizations. For example, Microsoft Research[1] developed a plug-in for Microsoft Word aimed towards the National Library of Medicine's PubMed Central[2] repository and allows submission of Word files only [Research 2011]. Another example is the Open Access Repository Junction (OA-RJ) aims at assisting publishers to deposit documents in multiple repositories with the use of a singular interface [EDiNA 2009]. Such applications do not cater for the complexity and variety of OER. Since the OER community is ever growing, applications need to aim towards helping wider audiences.

With that said, there is a need for the creation of more generic and simpler solutions to assist content creators, educators, authors and general users to deposit their content into repositories. Such solutions can also help the growth of OER repositories and the movement as a whole.

## 1.2 Aims

The main problem tackled by this research is investigating the possibility of a desktop solution to simplify the traditional repository deposit for OER. A repository system must accept deposits and represent OER correctly. Since this solution is developed for content creators, its usability and efficiency needs to be explored and identified. With this said, the specified questions this thesis aims to answer are:

[1] http://research.microsoft.com/
[2] http://www.ncbi.nlm.nih.gov/pmc/

1. How should a **repository system** be configured and adapted to be efficient for OER?

2. What is a suitable XML **metadata** representation for OER in a repository?

3. Is a development of a **desktop application** possible to ease the use of the repository for OER creators or endusers?

## 1.3 Methodology

The project integrates some mature and available technologies to simplify OER deposits to repositories. It was aimed to be seamless and familiar to authors and content creators. In other words, they should be able to integrate the project output application to their day to day work habits. This project started by exploring different OER repository structures available online. A set of exploratory interviews were conducted to get an idea of what some professionals in education know. The process was initiated by interviewing some lecturers and educational technology personnel. Specific requirements were collected and major issues were discovered at an early stage. The target population was identified and later liaised with in the design process. At this stage, several design considerations were decided upon, such as: the use of a standard digital library system as a repository that has been configured or modified as necessary to meet requirements; the support of a wide variety of digital objects; the use of a metadata standard that appropriately represents the digital objects; and the packaging of digital objects for their deposit.

A repository system was developed and adapted to handle OER. The necessary changes were identified including resource representation, metadata creation and item deposit process. The IMS Metadata Specifications [IMS 2006] for learning objects was implemented at this stage to represent the objects ingested into the repository. The reason behind using this metadata format is because it was specifically designed for learning objects [McClelland 2003]. A deposit service based on SWORD [Allinson *et al.* 2008b] was prepared for the repository for resource ingestion. Handlers for the support of additional file types were also added to the repository system.

After the repository was active the development of a desktop application commenced. A focus group qualitative design strategy was adopted to design an application prototype. Java technologies, such as the Drag and Drop data transfer features and Swing GUI widget toolkit, were used to implement the desktop application and its features. A SWORD [Allinson *et al.* 2008b] client was implemented for the desktop application to interact with the test repository. The desktop application was named *ORchiD*, an acronym for *Open educational Resources Depositor*.

The final system was evaluated for correctness and usability. The users were asked to test the deposit process and evaluate their experience with ORchiD, the metadata representation and the repository representation for their OER using an online survey.

## 1.4 Thesis Organization

**Chapter 2** presents an overview of concepts of OER and major projects in the area. Repository systems and best practice standards are discussed. An overview of the related work and systems is also presented and discussed.

**Chapter 3** discusses and presents the methodology to design and implement the solution.

**Chapter 4** presents the evaluation of the system as a whole. A user study design is explained and the results are presented.

**Chapter 5** provides the interpreted conclusions and presents the future work for enhancements and improvements.

Chapter 2

# Background and Related Work

## Contents



Open access and open resources have gained much attention from the world in the last few years. The interest in sharing information freely by the use of the World Wide Web has grown rapidly in many fields. Now, information is available in many different data forms - not only documents - because of the evolution in technology [Atkins *et al.* 2007].

## 2.1 Open Educational Resources

Open Educational Resources (OER) are teaching and learning materials, digital or digitized, that are shared freely online to the public. OERs include courses, documents, lectures, tests, images and video [Baker 2005].

Educational bodies - mostly higher education institutions - around the world have been using the various digital technologies to their advantage to create and disseminate their intellectual property to the public for many years now. Open Educational Resources (OER) have the potential to provide growth of an institution, internally and externally, in educational and learning capacities [Yuan *et al.* 2008].

### 2.1.1 The OER movement & Open Initiatives

Openness describes a resource or object being free for access and use with few conditions applied. The understanding of this concept has different forms but it is mainly the ability of use, reuse, share and remix resources [Tuomi 2006]. Many initiatives and institutions have contributed to this concept of freedom of knowledge. In this section some of the most popular are presented.

**Open Source Initiative (OSI)**

The OSI[1] was founded in 1998. The main aim of this initiative was to exploit the open source concept for the good of the community through its Open Source Software certification program. This allows free usage, distribution and modification of software adopting the license by any user. Software labeled with such a license, that is guaranteed by the Open Source Initiatives, complies with community standards.

**Open Content Initiative**

The Open Content Initiative[2], founded by David Wiley in 1998, was inspired by the success of the OSI. The initiative extends the concept of the OSI to learning content and educational resources. The basic and initial concept of the original license states that any object is freely available for use, modification, sharing or distribution with certain restrictions forced by a license.

**Open Access Initiatives**

There are many open access initiatives but three in particular serve the open access movement as milestones. The Budapest Open Access Initiative[3] (BOAI) was the outcome of the meeting hosted by the Open Society Institute in Budapest in December 2001. The initiative produced two strategies for open access: firstly, creation of open access journals; and secondly, self-archiving of scholarly work. The BOAI has been signed by just over five thousand individuals and five hundred organisations from around the world. Another meeting at the Howard Hughes Medical Institute in Maryland resulted in the Bethesda Statement on Open Access Publishing[4] in April 2003. A result of a conference at the Max Planck Society in Berlin in

[1] http://www.opensource.org/
[2] http://www.opencontent.org/
[3] http://www.soros.org/openaccess
[4] http://www.earlham.edu/ peters/fos/bethesda.htm

October 2003 is the Berlin Declaration on Open Access to Knowledge in the Sciences and Humanities[5], which encourages researchers to provide their resources on the Internet.

**Creative Commons**

In late 2002, Creative Commons released a set of open licenses to the public. These licenses are designed for digital content to help creators keep their rights and still declare their resources free to the public under certain conditions. ccLearn was launched in 2007 as the educational division of Creative Commons. The aim of this division is to use the potential of Internet technologies for open learning.

**Cape Town Open Education Declaration**

In September 2007, a meeting of open education activists from around the world[6] was organized by the Open Society Institute and the Shuttleworth Foundation. The meeting was entitled Open Sourcing Education and aimed at enhancing the efforts from around the world that support openness of educational resources and technology. The meeting involved a group of participants from academies, colleges, universities, institutes, foundations and other educational bodies that resulted in the production of the vision of the Cape Town Open Education Declaration. This declaration is meant to help the growth of the open education movement. Thousands of individuals and hundreds of organisations have signed this declaration.

Three strategies were presented in this meeting for the declaration to increase the efficiency and impact of OER. Firstly, educators and learners were to be encouraged to participate and be involved in the awareness and growth of the open education movement. This is to be done through producing and using open educational resource and encouraging others to do the same. Secondly, the release of intellectual property and resources are to be reviewed by the various producers to the public in formats accessible to all. Thirdly, opening education should be given a high priority.

### 2.1.2 OER projects

**MIT Open CourseWare**

In 2001, the Hewlett Foundation [Hewlett 2005] funded the Massachusetts Institute of Technology (MIT) in the launch of its OpenCourseWare initiative. The initiative is to make all of MIT's resources for undergraduate and graduate courses freely available on-line. Due to intensive media coverage of the MIT OCW launch, many institutes and universities were encouraged to share their resources online similarly [Abelson 2008]. Currently, more than 100 universities and institutions from around the world support the open publication of their materials.

**Connexions**

Complementary to the MIT OCW [Abelson 2008], the Connexions Project provides tools to help individuals create, publish, learn and build OERs. Connexions is an environment where users can work together to develop, share and publish

[5] http://oa.mpg.de/openaccess-berlin/berlindeclaration

[6] http://www.capetowndeclaration.org

scholarly content on the Web. The Connexions repository contains educational resources from a wide range of courses and subjects at most levels that are free to use and reuse [Atkins *et al.* 2007]. Another OER program is by Utah State University [Albright 2005], which provides open content and also open learning support through the Center for Open and Sustainable Learning (COSL). COSL has developed eduCommons, which is an OCW management system for others interested in initiating such a project in their institutions [Caswell *et al.* 2008].

In a general context, the mentioned projects present three OCW models [Wiley 2006] that could be adopted by other initiatives or projects.

## 2.2 Digital Library Systems

A digital library system is software that aids the creation, storage and management of digital collections. These systems are developed to manage various types of digital objects in many ways. A digital repository can hold a wide range of materials for a variety of purposes and users. It can support research, learning, and administrative processes. However, repository solutions are most viable and sustainable when they are built on open standards [Hayes 2005]. Open standards are publicly available descriptions of the ways in which systems can interoperate. Being publicly available, they enable developers to link together systems in innovative ways.

There are many Digital Library Systems used to create repositories. They have a wide range of features from extensibility and content specification to openness and interoperability. There are four widely used packages: Fedora [Lagoze *et al.* 2005], Dspace [Tansley *et al.* 2003], EPrints [Pinfield *et al.* 2002] and Greenstone [Witten *et al.* 2000]. All are open source and supported by different universities. The main issues here are flexibility, usability and interoperability with other applications.

### 2.2.1 Fedora

Fedora is an open source system for the storage, management and dissemination of different types of digital objects and their relationships. The key features of the Fedora repository architecture are, firstly, support of heterogeneous data types and adaptation to new ones, then the aggregation of mixed and possibly distributed data into complex objects and, also, the ability to specify multiple content disseminations of these objects [Staples *et al.* 2003]. Moreover, there is the ability to associate rights management schemes with these disseminations.

FEDORA's functionality [Lagoze *et al.* 2005] is broken into a set of services as follows; repository services are ways of depositing, storing and accessing data; index services are ways of discovering digital objects; collection services join digital objects and services into collections; naming services are for resolving then giving digital objects unique names; and, finally, user interface services provide users with interfaces to access the other services [Lagoze *et al.* 2005].

A digital object in Fedora comprises four parts: a unique persistent identifier; metadata required to facilitate the management of the object; a datastream, which is the digital content of an object like digital images; and, lastly, a disseminator that associates services with object. The Fedora Architecture is divided into four subsystems [Lagoze *et al.* 2005] and the Web services layer. The Core Subsystem layer includes the Management and Access subsystems. The management subsystem manages all operations on the digital objects. It also checks objects for validity and integrity when they are first deposited and then when they are changed. The management subsystem also is responsible for generating globally unique persistent identifiers for the objects. The Access subsystem implements the operations necessary for disseminating objects and discovering more information and behaviours for an object. The Security subsystem is responsible of enforcing the system's policies onto the objects and the users of the repository. The Storage layer, which comprises the storage subsystem, handles reading, writing and removal of data from the repository. Digital objects are stored as XML-encoded files. Finally, the Web Service Exposure Layer is where all the interaction with the users occurs.

### 2.2.2 DSpace

DSpace is designed to operate as a centralised, institutional service [Tansley *et al.* 2003]. The system is intended to reflect the flow of information in an institution. Parts of the system can be designated to the different communities of members within the institution. Web user interfaces are used by the members to deposit digital objects to the system. DSpace provides a platform to begin work on long term preservation strategies for digital material, including documents and other material used in scholarly research. A large number of universities and institutions around the world use this system [Tansley *et al.* 2003].

As for the functional features of DSpace [Smith *et al.* 2003], the system firstly defines a data model that reflects the basic data structure of an organisation. Descriptive, administrative and structural metadata is held for the archived content. The system also holds information on all users for authentication and authorisation to control accessibility and administration rights. Objects, relationships and metadata are checked before they are placed in the repository. A handle system generalises access and citation by the users. The system enables searching and browsing objects. It also supports the Open Archives Initiative Protocol for Metadata Harvesting. Finally, a Web user interface is provided for the access by end-users [Smith *et al.* 2003].

DSpace is implemented in Java and runs on any UNIX-like operating system such as Linux, as well as on Windows. DSpace also makes use of some other open-source systems like PostgreSQL for the relational database and Apache HTTPD server for certificate support. The system's architecture is divided into three layers. The system consists of storage, business logic and application layers [Tansley *et al.* 2003].The storage layer is responsible for the storage of the content and its metadata. The business logic layer manages the content of the repository, users, authorisation and workflow. The application layer consists of all the components and applications outside the DSpace installation that access the repository, like the Web user interface and the Open Archives Initiative Protocol for Metadata Harvesting service.

### 2.2.3 EPrints

EPrints [EPrints 2011b] is an open-source archive system based on a plug-in hierarchy structure. EPrints is aimed at producing open-access repositories. Its primarly used for institutional repositories and scientific journals but its base structure allows for archiving any filetype including audio, video, images and complex combinations. EPrints is based on a LAMP (Linux, Apache, MySQL and Perl/PHP/Python) architecture implemented in Perl.

EPrints core or main components consist of Datasets, a Data Storage layer and Utility methods. Datasets are the collections of objects in file storage. The Data Storage layer is the database and its controllers for distributed storage services. EPrints provide Utility methods and services including access, management, web publishing and dissemination [EPrints 2011a]. More details on EPrints are given in Chapter 3.

### 2.2.4 Greenstone

The New Zealand Digital Library Project at the University of Waikato produced a suite of software called Greenstone. Greenstone is an open-source digital library system for the construction and presentation of digital collections. It provides ways of structuring information or content and publishing it on the Internet or CD-ROM [Witten *et al.* 2000].

The architecture of Greenstone is based on two key components: the Receptionist and the Collection Server. The Receptionist provides the user interface, dispatching requests to the appropriate Collection Server or Servers and aggregating results for display back to the user. The Collection Servers provide abstract mechanisms to manage the contents of collections. The Receptionists communicate with Collection Servers using a defined protocol in the server configuration. In a default configuration the Receptionist and Collection Server are in a single executable and simple function calls are used for communication. This a single-server confguration called the null protocol. Greenstone has a Java-based client to support distributed environments. This client utilizes the Common Object Request Broker Architecture (CORBA) protocol. The collection service uses two database systems: MG (Managing Gigabytes), used for full-text search and retrieval; and GDBM (GNU Database Manager), used for the collection information database. Greenstone attempts extensibility of its architecture through Plug-ins and Classifiers [Buchanan *et al.* 2005].

### 2.2.5 Comparisons of Digital Library Systems

The literature [Pyrounakis & Nikolaidou 2009, Warr & Hangsing 2009, Lihitkar & Lihitkar 2012] suggests that several groups have evaluated, analyzed and compared digital library systems to help in the decision making of selecting the appropriate software for a specific set of organizational requirements.

A study in Greece [Pyrounakis & Nikolaidou 2009] attempted to compare the latest releases of these Digital Library systems: Fedora, DSpace, Eprints, Greenstone and Keyspace. The comparison was based on stated characteristics to produce a set of guidelines or cases which each of the systems is suitable for. In the paper, the authors begin by stating that the advantage of having many DL software systems becomes an obstacle when selecting one for an organization. Ten characteristics of DL systems were presented for comparison which were based on internal structure, usability, flexibility, services and support of the DL system. The study concluded that it is difficult to propose one system that is suitable for all cases due to the various requirements organizations may have. Even though, four cases were presented where an organization or institution needs a DL system for: *Case 1*, containing research papers and dissertations produced by students and staff; *Case 2*, publishing digital content in a simple form within strict time limits plus integrating it with a portal like a website; *Case 3*, hosting and preserving digitized collections from libraries, archives and museums and *Case 4*, publishing electronic books in an easy to use customizable system. DSpace was proposed as being the most suitable choice for *Case 1* due to its default representation for communities like university departments and collections like research papers and dissertations. For *Case 2*, the authors suggested that Keystone or EPrints could be best since their presentation and storage are separate while not being bound to any specific metadata standard and also, provide simple web interfaces for submission and content presentation. As for *Case 3*, Fedora was recommended because preservation, use of multiple metadata standards and different formats of content are the highest priority needs. As the authors claim, while Fedora provides a very customizable modular architecture without easy web interfaces or built-in functionality is still the best choice where hosting many collections and different digital materials is a requirement. Greenstone was proposed for *Case 4* because of its easy heirarchical representation and full text search capability.

A similar comparative but rather comprehensive paper [Warr & Hangsing 2009] introduced digital libraries and presented a comparative analysis of DSpace, Greenstone, EPrints and Fedora. Features common to the software were selected for the analysis. The main objective of the study was to analyze and identify the strengths and weaknesses of the mentioned systems. These features included: content management and administration; interoperability and compliance with standards; types of contents and organizations; total number of installations; user interfaces and content retrieval features. It was found that all systems lack certain functions that were perceived to be important by the authors. According to this analysis, DSpace was able to fulfill most of the selected features and described as having an excellent workflow process unlike Greenstone having no built-in workflow process. The authors also concluded that EPrints is a powerful system for opening access to scholary. Finally, Fedora's key strength was identified as preservation standardization support through multiple versions and formats of digital objects.

It is worth noting that most of these software systems are continuously changing and growing, in terms of new features and services. As difficult as the comparison may be, there is yet no standard process or criteria to evaluate these systems.

A more recent study [Lihitkar & Lihitkar 2012] compares recent versions of a larger set of open source digital library systems and ranks them according to divised scores in a criteria of features, functions and usability aspects. The authors state that the ten systems investigated, including the aformentioned, share the advantage of being flexible to be customized as much as needed in most cases. All systems also need prerequisite software like database, programming environments or Web server. EPrints was found to need the most number of installation prerequisites but works under all operating systems. It was also concluded that Greenstone and DSpace need the expert knowledge of XML or HTML to work with metadata in store unlike the other systems. Generally, the three mentioned systems here were graded as excellent with high scores. The authors recommended Greenstone or DSpace as the choice of software for libraries.

It is clear from the literature that it is difficult to make an ultimate selection when choosing a digital repository software to use. Similar desirable traits can be found among the packages. Furthermore, the continuous growth of such systems just makes it that much harder. Much can be learned from reported experiences of the software usage and comparisons made.

## 2.3 Metadata schemas & application profiles

The concept of metadata has been defined in many ways. Metadata can simply be understood as information about information [Anido 2006]. It was also stated that metadata is any data which conveys knowledge about an item without requiring examination of the item itself [Haase 2004]. Metadata record consists of structured information about the resource it describes. Due to this structure, metadata facilitates the discovery, managment and retrieval of that resource. In general, metadata schemas facilitate the description of the content, quality, condition, authorship and any other characteristics of some objects or data [Al-Khalifa & Davis 2006]. Metadata specified for educational purposes extend the scope of the description that can be included in metadata records with information that has particular educational relevance [Hatala *et al.* 2004b].

Metadata specifications provide a structure that describes a resource and specifies how it is used [IMS 2006]. Authors or developers of OER have a choice of a variety of tools from a range of vendors to produce their learning material and electronic resources. Due to that, many formats can be used to represent the same information content. This results in the complexity of specifying one metadata representation standard that suits all of these formats and types of learning objects.

Dublin Core (DC) Metadata [Weibel 2010] and Learning Object Metadata (LOM) [Barkman *et al.* 2002] are two popular standards that facilitate cataloging, searching and reuse of digital objects such as OER [McClelland 2003]. The main difference between the two is that LOM was originally developed specifically for the domain of education and training while the DC Metadata Element Set (DCMES) was originally developed for general resources. LOM is more popular due to its wide acceptance in learning communities. The metadata specification used to represent the objects in a repository has an effect on the performance and correctness of the repository and hence even the usability and the interoperability with other applications.

Application profiles are metadata sets specified for a particular application. They can be subsets of a standard or a collection from different standards to optimize their usage for that particular application [Heery & Patel 2000].The rest of this section describes and discusses these standards further, draws simple examples of XML representations and presents some application profiles. The standards and specifications discussed here are not complete but summarized for the purpose of this project.

### 2.3.1 Dublin Core

DC was developed to address the need to improve retrieval of information resources. It is a generic and broad metadata set intended to support description of any online resource. It was deliberately limited to only fifteen elements that have applicability over a wide range of information resources [Cathro 2009].

#### 2.3.1.1 DC Metadata set

The specification for the DC Metadata Element set describes the fifteen elements. The Dublic Core Metadata Initiative (DCMI)[7] maintains DC and describes a detailed set with metadata vocabularies and technical specifications like namespaces and recommendations of use. Table 2.1 lists the DC elements summarized and derived from the specification [Kunze & Baker 2007].

[7] http://dublincore.org/documents/dces/

| Element Name | Description |
|---|---|
| Title | the name given to the resource |
| Creator | the name of the entity primarily responsible for making the resource like a person or organization |
| Subject | the topic of the resource like keywords or classification codes |
| Description | an account of the resource like an abstract or table of contents |
| Publisher | the name of the entity responsible for making the resource available |
| Contributor | the name of the entity responsible for making contributions to the resource |
| 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 |
| Format | the file format, physical medium or dimensions of the resource |
| Identifier | an unambiguous reference to the resource within a given context |
| Source | a related resource from which the described resource is derived |
| Language | a language of the resource |
| Relation | a related resource |
| Coverage | the spatial or temporal topic of the resource or the jurisdiction under which the resource is relevant |
| Rights | information about rights held in and over the resource |

Table 2.1: Dublin Core element set

A simple DC XML example describing a book is presented in the following box. The example shows the simplicity of the metadata structure.

```
<title>XML Visual Quickstart Guide 2nd Edition</title>
<creator>Kevin Howard Goldberg</creator>
<format>Book</format>
<identifier>ISBN 978-0321559678</identifier>
```

#### 2.3.1.2 DC application profiles for education

DCMI identified some limitations to the basic set of fifteen elements and developed qualifiers to add further details to make the descriptions richer to the users [Guha 2008]. Increasing interoperability among applications was a main goal. The qualifiers are specified as "*Element refinement*" and "*Encoding schemes*" [DCMI 2005]. The Element refinement qualifiers enhance the meaning of an element by making it more specific like submitted date or published date for the unqualified

| Element Name | Description |
|---|---|
| Audience | A category of user for whom the resource is intended |
| Audience.Mediator | An entity that mediates access to the resource |
| Standards | A reference to an established education or training standard to which the resource is associated |
| Standards.Identifier | Where available, an identifier that serves to uniquely identify the standard being associated |
| Standards.Version | Information identifying the version of the standard being referenced |
| InteractivityType | The flow of interaction between this resource and the intended user |
| interactivityLevel | The degree of interactivity between the end user and this resource |
| typicalLearningTime | Approximate or typical time it takes to work with this resource |

Table 2.2: data elements extending DC to make up the DCMI-EMS

element `Date`. Encoding Schemes identify and specify controlled vocabularies or value sets for elements. For example a specified format for the value of the element Date or a controlled vocabulary of choices described by the element `Subject`.

DCMI Education Metadata Set (**DCMI-EMS**) or DC Education Schema was created as a DC application profile for describing educational resources with a focus on five particular areas of interest to educational metadata projects: users; duration; learning processes; standards and quality [Heery & Patel 2000]. In addition to the basic DC element set DCMI-EMS specifies two more fields; *Standards* and *Audience*. Three fields from LOM were also added to the specification as shown on Table 2.2.

Education Network Australia (EdNA) initiative produced a DC metadata application profile for its Directory Service [Mason & Ip 1998]. The EdNA profile includes six additional elements to meet the specific application to the Australian education domain. The specification adds more elements for the description of resources. Gateway to Educational Materials (GEM) is also another example of a project that extended DC and developed a schema with 7 additional metadata elements to meet specific educational needs [Greenberg 2005].

### 2.3.2 Learning Object Metadata

The International Metadata Standards (IMS) Global Learning Consortium, in collaboration with other organisations, developed the IEEE LOM specifically to represent educational resources [Roy *et al.* 2010]. LOM is the most comprehensive XML metadata specification for learning objects. Its data model shapes into a hierarchy of 78 elements. LOM comprises nine main elements that contain sub-elements. The sub-elements can either hold the data or contain other sub-elements themselves. The nine elements describe characteristic groups of learning objects specified as: `General`, `Life cycle`, `Meta-metadata`, `Educational`, `Technical`, `Rights`, `Relation`, `Annotation` and `Classification` categories.

#### 2.3.2.1 LOM set

In Figure 2.1, the elements specified in the Draft Standard for LOM [Barkman *et al.* 2002] are depicted. Each element has a *value space* and a *datatype*. The *value space* describes restrictions on the data entered for the element and the data type specifies the type or a set of values that can represent the element. For example, the element `General`, shown in Figure 2.1, has sub-elements *Title*, *Language*, *Description*, *Keyword*, *Coverage*, *Structure* and *Aggregation level*. The sub-element *Identifier* can have multiple values in this element whereas there can be only one `Catalog` and `Entry` for each *Identifier* element. In other words, the scenario of having a learning object with multiple catalog entries is describable. *Langstring* is the datatype of the sub-element *Title*, which is basically a set of characters with a *language* attribute. When adopting LOM in a specific implementation it is not necessary to use all data elements.

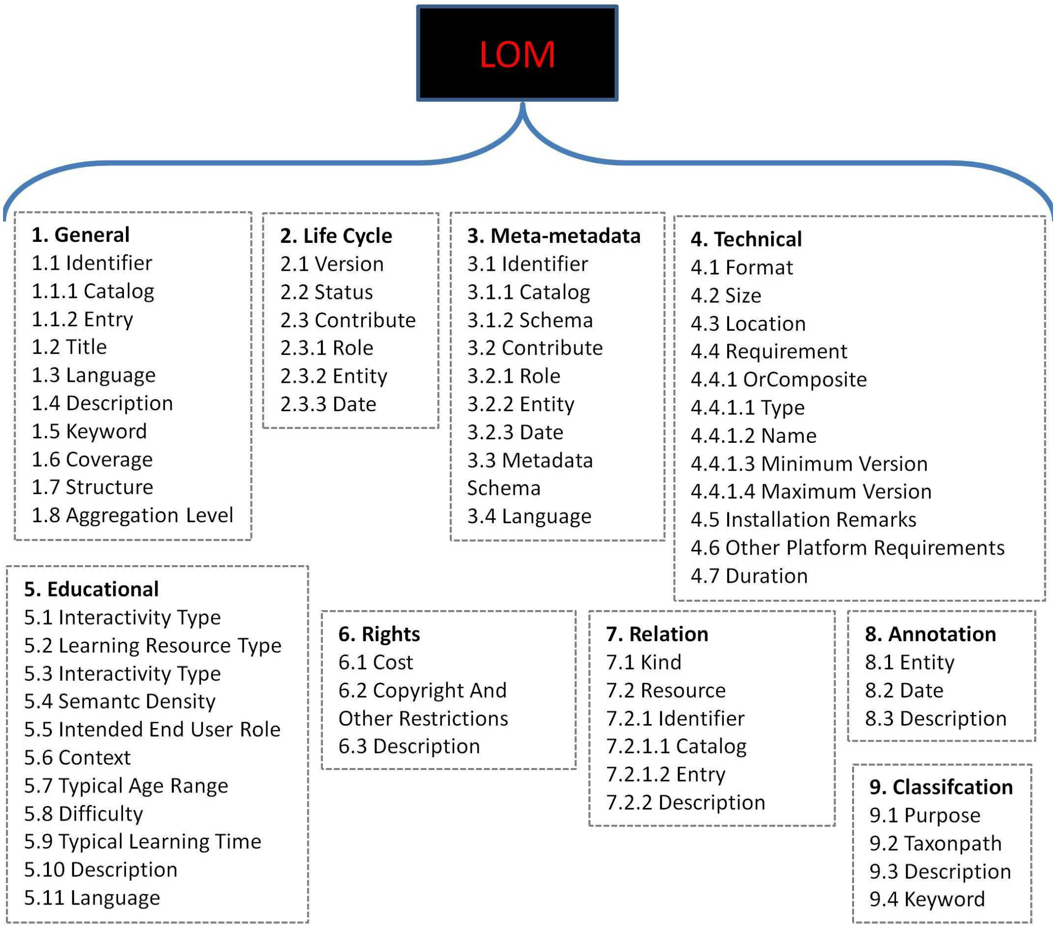



Figure 2.1: Base Learning Object Metadata Set

The properties associated with each element are derived from the IEEE 1484.12.1-2002 Draft Standard for LOM [Barkman *et al.* 2002] and summarized in Table 2.3. The numbers refer to the elements that are stated in the specification and shown in Figure 2.1.

| Property | Data elements |
|---|---|
| has sub-elements | 1, 1.1, 2, 2.3, 3, 3.1, 3.2, 4, 4.4, 4.4.1.1, 5, 6, 7, 7.2, 7.2.1, 8, 9, 9.2, 9.2.2 |
| can have multiple values per instance | 1.1, 1.3, 1.4, 1.5, 1.6, 2.3, 2.3.2, 3.1, 3.2, 3.2.2, 3.3, 4.3, 4.4, 5, 5.2, 5.5, 5.6, 5.7, 5.10, 5.11, 7, 7.2.1, 8, 9, 9.2, 9.2.2, 9.4 |
| order of values is significant | 2.3, 2.3.2, 3.2, 3.2.2, 4.3, 5.2, 9.2.2, 9.4 |
| has a controlled value space | 1.1.1, 1.1.2, 1.3, 1.7, 1.8, 2.2, 2.3.1, 3.1.1, 3.1.2, 3.2.1, 3.2.2, 3.3, 4.1, 4.2, 4.4.1.1, 4.4.1.2, 4.4.1.3, 4.4.1.4, 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.8, 5.11, 6.1, 6.2, 7.1, 7.2.1.1, 7.2.1.2, 8.1, 9.1, 9.2.1, 9.2.2.1 |
| takes a LangString value | 1.2, 1.4, 1.5, 1.6, 2.1, 4.5, 4.6, 5.7, 5.10, 6.3, 8.3, 9.2.1, 9.2.2.2, 9.3, 9.4 |
| takes a Datetime value | 2.3.3, 3.2.3, 8.2 |
| takes a Duration value | 4.7, 5.9 |
| takes a CharacterString | 1.1.1, 1.1.2, 1.3, 2.3.2, 3.1.1, 3.1.2, 3.2.2. 3.3, 3.4, 4.1, 4.2, 4.3, 4.4.1.3, 4.4.1.4, 5.11, 7.2.1.1, 7.2.1.2, 8.1, 9.2.2.1 |

Table 2.3: Properties of the data elements

The box below contains a sample LOM XML record. This is only a section of what a full XML representation could look like but it protrays the increasing complexity when directly compared to DC XML.

```
<lom>
<general>
    <title><langstring>developerWorks : XML</langstring></title>
    <description><langstring> The XML zone on the developerWorks Web site is
designed for developers. You'll find tools, samples, standards information,
education, news and events, and links to XML community forums and Web sites.
</langstring> </description>
    <keyword><langstring>xml resources</langstring></keyword>
    <keyword><langstring>xml programming</langstring></keyword>
</general>
<technical>
    <format>text/html</format>
    <location>http://www-106.ibm.com/developerworks/xml/</location>
</technical>
<educational>
    <learningResourceType><source>DCMIType</source><value>Text</value>
    </learningResourceType>
</educational>
</lom>
```

#### 2.3.2.2 LOM application profiles

LOM application profiles are mostly specifications of the usage of the elements already specified, with a concentration on the vocabularies provided.

IMS uses the LOM as a basis for its metadata specification, IMS Learning Resource Metadata (LRM) Information Model [IMS 2003d]. IMS has also contributed to LOM by introducing best practice guides for metadata implementers and an XML binding specification. The current IMS specification consists of all 76 LOM elements [IMS 2006].

The UK LOM Core is an application profile of the IEEE 1484.12.1-2002 Standard for LOM that has been optimised for use within the context of UK education. The specification provides guidelines on all LOM elements, recommendations on usage and defines UK vocabularies. The UK LOM Core does not specify any omissions or complete changes in the basic LOM definition but states whether an element should be mandatory or optional in adoption and provides additional information on use [Duval *et al.* 2006]. Similar to the UK LOM is the Canadian Core Metadata Application Profile specified for Canadian repositories with the aim of simplifying the LOM metadata and resolving some ambiguities in the initial draft [Friesen *et al.* 2002]. Another LOM application profile is the Sharable Content Object Reference Model (SCORM) which adapted specifications from different organisations and provided a collection of documents attempting interoperability, accessibility and reusability of learning content [Roy *et al.* 2010].

## 2.4 Content packaging

A Content Package is a file containing resources and associated metadata. The main aim behind Content Packaging (CP) is preparing content for transport between systems [Lukasiak *et al.* 2004]. A number of specifications have been provided by different organizations for achieving this. These structures provide the basis for standardized data bindings that allow the software developers and the implementers to create instructional materials that are interoperable across authoring tools, learning management systems and run time environments [Sierra *et al.* 2005]. This section discuss some of these standards.

### 2.4.1 IMS Content Packaging

The IMS CP specification is consists of three documents: the Information Model describes the logical structure of a content package [IMS 2003b]; the XML Binding describes the components and organization of a CP in logical terms [IMS 2003c] and the Best Practice Guide describes the use of the CP in learning management packages with examples [IMS 2003a].

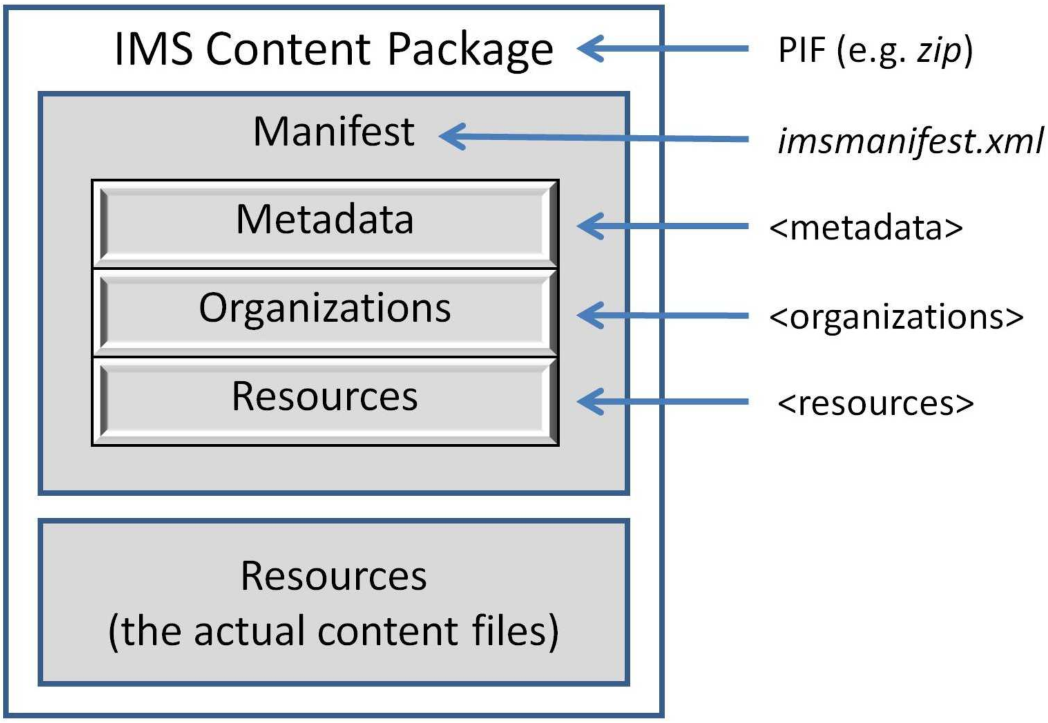



Figure 2.2: IMS Content Package assembly

Figure 2.2 shows the components of an IMS CP as specified by the standard. The CP contains a single required Manifest file and the resource(s) it describes. The manifest file must be named *imsmanifest.xml* and be placed at the root of the Package Interchange File, which is recommended to be of the type *zip*. The Manifest file is expressed in XML for creating the data structure and divided into sections. Typically, it should include: `Metadata` in a specified standard describing the resources; `Organizations` for how the physical files included in the package need to be arranged for recieving the intended experience and `Resources` referencing the actual deliverable content.

### 2.4.2 Other specifications

Metadata Encoding and Transmission Standard (METS) [Lukasiak *et al.* 2004], Moving Pictures Experts Group (MPEG21) [Bekaert 2003] and SCORM [Learning 2011] also specified mechanisms for packaging resources like learning objects. The main principles behind packaging standards are similar.

SCORM is a set of technical standards and specifications from the Advance Distributed Learning (ADL) initiative that aim to regulate the the development, packaging and delivery of content [Learning 2011]. In SCORM, content consists of a set of reusable learning objects refered to as Sharable Content Objects (SCO). SCOs are combined with delivery instructions and metadata in a XML manifest file to make up the package. The package also specifies the API needed to use the content and the sequencing instructional information of how the content is to be used. SCORM packages are not changeable once they are deployed and has limited support for modern eductional technologies like simulations, wikis and Web-based learning environments because it was created for personal learning and training [Kavcic 2011].

## 2.5 Deposit and publishing protocols

Repositories ingest new content either through their own user interface or through a Web service [Tansley *et al.* 2003]. The main aim of depositing and publishing protocols is to facilitate the transport of resources or metadata between systems that create or generate content and systems that manage, publish or deliver content. These systems include repositories, learning and content management systems, third-party applications, digital libraries and similar systems. These protocols are typically implemented as interoperability services by the systems housing the resources. Some of these protocols or specifications are explained further in this section with a larger focus on some for their relevance to this project. Sections 2.5.2 and 2.5.1 present and discuss the Simple Publishing Interface (SPI) and Simple Web-Offering Repository Deposit (SWORD) depositing specifications with a focus on the technical issues. Following that, section 2.5.3 discusses more deposit or publishing protocols. A few projects that have made use of such specifications or standards are mentioned throughout this section.

### 2.5.1 Simple Web-service Offering Repository Deposit

The SWORD project [Allinson *et al.* 2008b] was funded by the Joint Information Systems Committee (JISC) to improve the efficiency of repository deposit and produce an interoperable standard. The project implemented interfaces based on this protocol on four major repository systems: EPrints, DSpace, Fedora and IntraLibrary. Example clients and SWORD Java API was also produced. The SWORD profile has also been used by a range of systems [Lewis *et al.* 2012].

The SWORD project is a profile of the Atom Publishing Protocol (AtomPub). AtomPub [Gregorio & De hOra 2007] is a protocol specified for publishing and editing Web resources at the application level. AtomPub is a widely adopted standard for Web feeds in blogging and websites with regularly changing content. The protocol uses HTTP [Fielding *et al.* 1997] for basic transport of Atom-formatted representations. The Atom format is an XML language described in the Atom Syndication Format specification [Nottingham & Sayre 2005]. The SWORD profile adds to Atom the creation of compound resources like archive files, support for mediated deposit on behalf of another client or user and relaxing the deposit process to support server specified workflows.

SWORD specifies client and server implementations. A SWORD-compliant server at a repository could offer a SWORD deposit service for a range of clients, like authoring tools or data generating machines. A standard deposit interface can offer many functionalities, like deposit from multiple locations, not just a traditional deposit interface on the repository's website. A server can specify support for formats, mediated deposit, collections or deposit locations and file types. SWORD also offers some developer features attempting to lower cost of implementation and configuration. These include logging and deposit testing features [Lewis *et al.* 2009].

SWORD servers in repositories specify and define the deposit service in a service document. The service document is retrievable by a client using an HTTP GET. This is offered as a URI. The service document is client-specific and informs the client of what the repository is offering. Figure 2.3 shows an example of a service document. As defined in the AtomPub specification [Gregorio & De hOra 2007], a service document "*describes the location and capabilities of one or more Collections, grouped into Workspaces*". A collection is a set of resources grouped together in a workspace. This is clarified in the example of a Service Document in 2.3.

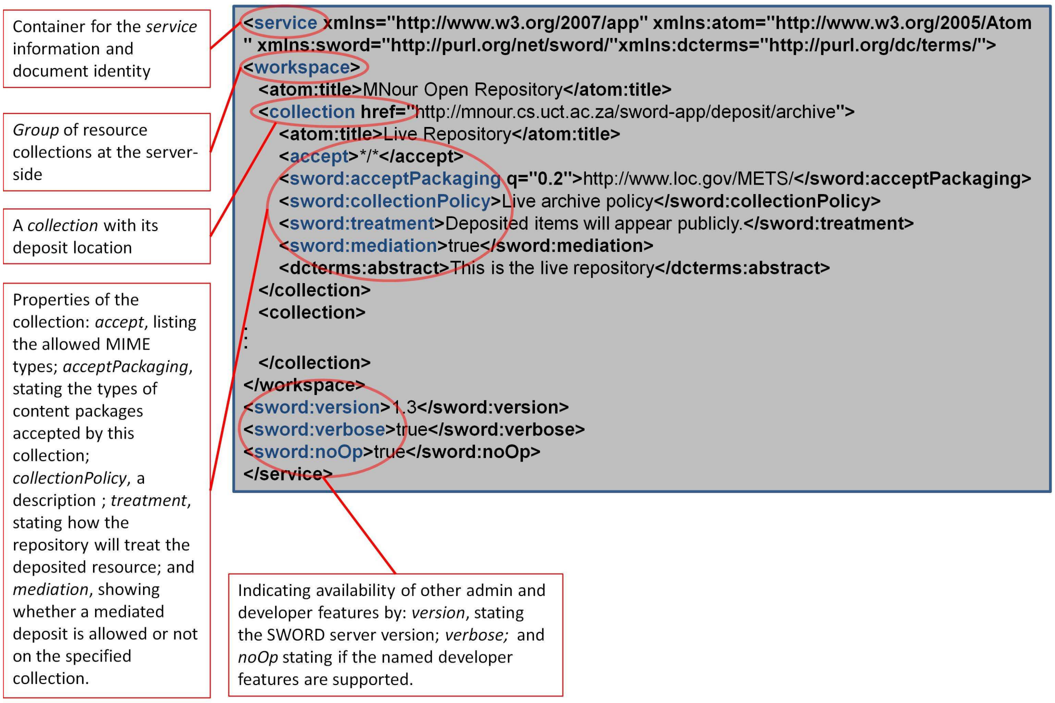



Figure 2.3: An annotated example of a SWORD service document

A basic deposit process using SWORD [Allinson *et al.* 2008a] is shown in Figure 2.4. The steps are described as follows:

1. The client requests a service document using a URI specified by a repository. An HTTP GET method is used and the user credentials may be included in the HTTP header. The service document tells the client how to deposit into the repository.

2. The server checks the user credentials for the usage of the sword interface and returns either the service document on a successful authentication or an error document in case of failure.

3. The client finds a deposit location in the service document and requests a deposit by sending the resources and metadata in a package.

4. The server perform checks specified by the repository and tries to deposit the resources, then returns a response to the client.

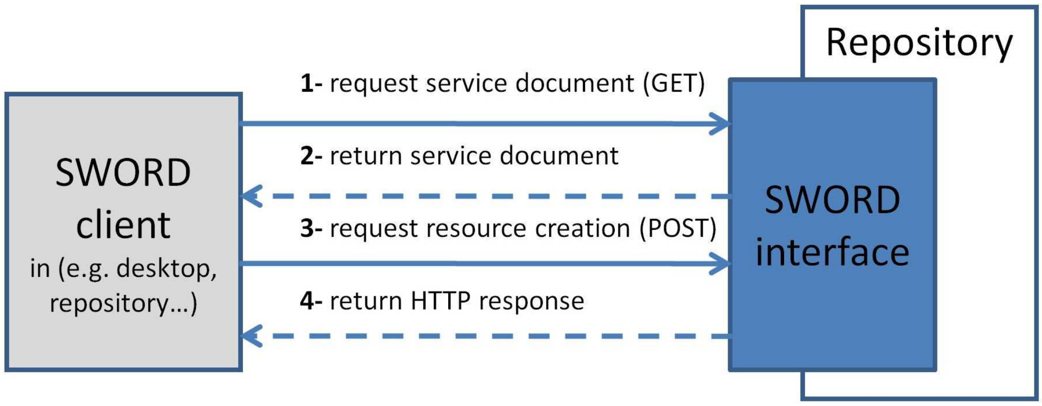



Figure 2.4: SWORD deposit client and server interaction

SWORDv1 only supports the deposit of items and metadata into repositories. Due to this limitation, SWORDv2 was released and accommodates further managment features like editing, updating and deleting resources [Jones 2011].

### 2.5.2 Simple Publishing Interface

The Simple Publishing Interface (SPI) was developed in the European Committee for Standardization (CEN) workshop on learning technologies [Ternier *et al.* 2008]. SPI was aimed towards facilitating communication between content producing tools and repositories. The main focus of the specification was depositing resources or metadata into repositories and enabling interoperability among repositories. SPI uses *source* and *target* as notations to differentiate between systems that issue requests and systems to which publication requests are sent. These terms are used here to discuss the SPI architecture.

SPI consider some scenarios for communication between source and target as shown in Figure 2.5. Firstly direct deposit of metadata and/or resources from *source* to *target* where a source is a learning management system like Moodle [Dougiamas & Taylor 2003] or an authoring tool like a word processor and the target is a repository. Secondly a similar scenario except with a middleware application where some processing like automatic metadata generation or other service is performed prior to forwarding to the repository [SPI 2010]. In this second scenario, the middleware application firstly acts as a target then as a source. An example of such an architecture is the Abstract Learning Object Content Model (ALOCOM) [Verbert & Duval 2007]. ALOCOM produced a Microsoft Powerpoint[8] plugin that sends slides to a middleware that disaggregates or breaks up the slides into smaller re-usable components and automatically generates metadata for them before sending them to the ALOCOM repository. As described in the scenario, the middleware acts as a *target* for the ALOCOM powerpoint plug-in and a *source* for the repository.

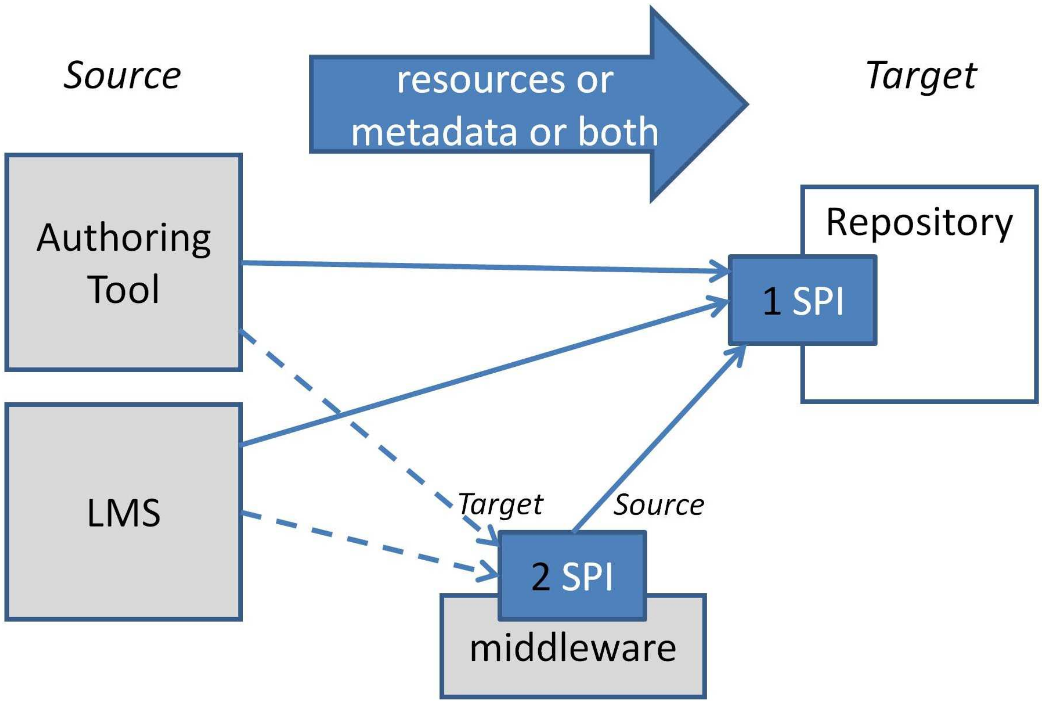



Figure 2.5: SPI design scenario examples

SPI specifies several features and methods for implementations to offer. The Learning Resource Exchange (LRE) is an example of a service that implemented SPI to extend its features [SPI 2010]. LRE [Massart 2007] helps teachers and schools find educational resources from different providers and countries. Initially, OAI PMH was the main mechanism for collection of metadata by the LRE. OAI PMH is a metadata harvesting protocol and is implemented as a service at the repository's side. Some of the content providers refused to issue OAI targets for security concerns and hence the SPI service was used. The difference is that SPI acts as a "push" mechanism rather than a "pull" mechanism like the OAI PMH. In other words, the content provider initiates the process in this scenario [Ternier *et al.* 2010].

[8] http://office.microsoft.com/en-us/powerpoint/

The main difference between SPI and SWORD specifications is that SWORD deals with metadata by packaging it with a resource, unlike SPI that treats metadata and resources as distinct while linking them with identifiers. SWORD is also a profile of AtomPub while SPI specifies a binding mechanism for AtomPub.

### 2.5.3 Other Deposit protocols

The Search/Retrieval via URL protocol (SRU) offers a Record Update service [Morgan 2004] that allows for remotely creating, replacing and deleting of metadata records within a compliant database. This specification does not deal with publishing resources or content. For this reason this protocol can be implemented for referatories that house links to resources and point to where resources are published.

Aviation Industry CBT Committee (AICC) specified the Package Exchange Notification Services (PENS) protocol that provides a notification service for content packages [AICC 2006]. PENS does not support publishing metadata instances or sending content across. Instead, a source can announce the location of a package that is available for transport. A PENS notification informs a system of the availability of a new resource by including the URL for the resource in a message. The system receiving the notification can then retrieve that package.

An implementation of messages proposed in IMS Digital Repositeries Interoperability specification [IMS 2003d] is given by the EduSource Communication Layer (ECL) [Hatala *et al.* 2004a]. ECL implements requests for submitting learning objects or metadata to a repository using messages. ECL builds on IEEE LOM metadata only and does not include parameters or methods to set different metadata schemas or even an application profile of IEEE LOM. ECL does not specify any distinction between sending metadata or resources.

A publishing specification was implemented in Ariadne[9] Web magazine [Ternier & Duval 2003]. Knowledge Pool System (KPS), a Web Services based approach was introduced that facilitates integrating the publishing process into applications where learning objects are either consumed or produced. KPS specifies an InsertService that defines a document or metadata ingestion method into Ariadne using SOAP[10] as a communication protocol. A version of this API was implemented by the ProLearn project [11] for other organisations like MACE and TENCompetence [Prause *et al.* 2007].

The Open Knowledge Initiative (OKI) [Hatala *et al.* 2004c] has specified Open Service Interface Definitions (OSID) in order to simplify and enhance the development of educational applications. More specific to content deposit is the Repository OSID that defines an Asset interface that manages both content and metadata records. An Asset is specified as a digital object or resource like a document. The Repository interface offers methods for adding and deleting records to an Asset. The asset interface also defines methods to associate content to an asset.

[9] http://www.ariadne.ac.uk/
[10] http://www.w3.org/TR/soap/
[11] http://www.prolearn-project.org/

## 2.6 Related Work

In this section several projects that focus on or offer resource deposit to repositories or other systems are presented and discussed.

### 2.6.1 Repository deposit solutions

**EM-Loader**

Extracting Metadata to Load for Open Access Deposit (EM-Loader) project was introduced to address the complaints from users that repository deposits are time consuming for self-archiving. The project states that most academics need to maintain professional personal websites listing and linking to their publications. The main issue is adding bibliographic metadata that is needed for submission of papers to repositories. The project aims to reduce this effort by linking a system designed for publication list management on a webpage. The API introduces automated interfaces including SWORD for deposit [Howell & Stuart 2009].

**Authoring Add-in for Word**

Microsoft Research[12] developed a plug-in for Microsoft Word to improve resource discovery and publication by integrating the writing process and metadata associations. The work was aimed towards the National Library of Science's PubMed Central[13] repository. The add-in or plug-in allows users to create and manipulate metadata in the National Library of Science's XML format. This project uses the SWORD protocol and allows users to deposit word documents to SWORD-compliant document repositories [Research 2011].

**DepositMO**

The Modus Operandi for Repository Deposits (DepositMO) project aims to extend SWORD to enable features such as resource discovery and synchronisation. The project is developing tools for Word 2010 to deposit directly to repositories and for desktop management systems to drag and drop into a folder that instantly synchronizes with the repository location setup by the user. At the time of this thesis, this project developed demonstration clients and is still at an early stage [Tarrant *et al.* 2012].

**OA-RJ**

The Open Access Repository Junction (OA-RJ) project aims to simplify deposit of resources from publisher to repository. The main concept is to allow publishers to deposit into multiple repositories at once instead of performing one on one deposits to various repositories. At the same time, keeping one to one relationsips between publisher and repositories. In other words, the publisher works with one interface, the OA-RJ, instead of interacting with varying repository interfaces that usually have different requirements. With that said, this should minimise relationships controlled by either publisher or repository [EDiNA 2009].

**EasyDeposit**

[12] http://research.microsoft.com/
[13] http://www.ncbi.nlm.nih.gov/pmc/

A custom Web-based deposit interface was developed for the University of Auckland Library to assist students in depositing theses to the institutional repository. This client, EasyDeposit, uses SWORD to perform the deposit. It also allows the user to select a Creative Commons license. A library catatalоguing team is required to complete the metadata associated with each thesis at some stage after a thesis is deposited [Lewis 2011].

**DURA**

Direct User Repository Access (DURA) focuses on integrating the institutional deposit process with the researcher's workflow. DURA specifies metadata collection techniques by integrating systems like Mendeley with the institutional repository [Wells 2010].

### 2.6.2 Other Systems and Projects

**BioMed Central automated deposit via SWORD**

An experiment conducted by the joint collaboration of the MIT and BioMed Central explored SWORD automatic deposits of BioMed publications into the MIT's DSpace institutional repository [Duranceau & Rodgers 2010] A nightly automatic deposit was agreed upon and a crosswalk between BioMed's METS metadata and MIT's DC was investigated. The publications had to meet MIT's specifications for publications to be accepted into the repository. The authors of the article reported a 36% rejection due to challenges faced. Nevertheless, a potential gain on time saving, deposit activity and speed was noted as pointed out in the report.

**Kepler**

Kepler has an architecture comprising of utilities that allow production of a digital library. It also provides control to the user of his/her resources in a personal digital library that is backed with larger organisational digital libraries [Maly *et al.* 2004b].

Kepler includes a structured API [Maly *et al.* 2004a] that defines various functions of different modules. A metadata manager module allows support of desired metadata formats for the system through metadata driver modules. The metadata manager also implements the OAI-PMH API that communicates with a Driver Manager. The Driver Manager is responsible for the user interfaces and also allows users to interact with the system. Kepler includes a validation module that validates the metadata at the publication stage. Server-side architecture is also implemented to allow access from anywhere with Internet access availability [Maly *et al.* 2004a]. Moreover, the system provides a variety of other services like information exchange between libraries and search through a group server.

**OER Publishing API**

The main goal behind the OER Publishing API (OERPub) is to simplify the publishing process specifically for open education resources that are adaptable and remixable. The API is a profile of the SWORD protocol. It specifies "a SWORD service that operates within an editing environment that publishes to a public repository of published content" [Fletcher 2012]. A tool was built using this API to deposit documents to Connexions. The tool is called the Connexions Document Importer[14].

**Blackboard Learn**

The Blackboard Learning System is a widely used virtual learning environment and course management system developed by Blackboard Incorporation [Bradford *et al.* 2007]. Recently, Blackboard Learn has released new intiatives for the publishing, sharing and consumption of OER within its systems. This is to assist lecturers or content creators to publish their resources under Creative Commons open licenses using Blackboard. The name of this free version is CourseSites[15].

## 2.7 Summary

Many projects and tools have emerged in the past decade that can be used to assist in the OER movement. Interoperability issues in content delivery and transfer has caused the development and growth of many standards focusing on different issues. Open access repository systems were created to archive, preserve, access and disseminate digital resources. As the growth and need for better solutions to assist in publishing resources grew, several solutions and projects were initiated to simplify the traditional repository deposit. Specific deposit protocols were developed and adopted by applications.

Chapter 3 discusses the integration of some of these technologies to develop the Open education Resources Depositor (ORchiD).

[14] http://oerpub.org

[15] http://softchalk.com/2011/05/put-your-courses-online-for-free-with-coursesites-by-blackboard

Chapter 3

# Design and Implementation

## Contents



The main aim of this project is the simplification of the repository deposit process. The system design was based on a number of design considerations and goals, such as: the use of a standard digital library system as a repository that has been configured or modified as necessary to meet requirements; the support of a wide variety of digital objects; the use of a metadata standard that appropriately represents the digital objects; and the packaging of digital objects for their deposit. To reach these goals, a desktop application was designed and implemented to help users ingest their educational content into an online repository in a seamless manner. The basic idea is to keep the direct user interaction with the repository to a minimum while completing the desired task.

Before the design of the application interviews with some of the educational technology staff at the University were conducted to collect requirements and confirm the need for easier solutions for OER. A focus group was also conducted to design an initial prototype of the desktop application. In this chapter, these are described before a discussion of the existing tools and newly developed components, integrated to form the system, is presented.

## 3.1 Initial Interviews and System design

This research was conducted at the Digital Libraries Laboratory in the Department of Computer Science at the University of Cape Town. Unstructured interviews were conducted with individuals from some departments and offices to collect information and gain awareness of what the educators know about OER. In-depth or unstructured interviewing is a data collection technique used to elicit information to achieve an understanding of the interview's point of view or explore interesting areas for further investigation [Berry 1999]. The interviews were mostly informal and included many open ended questions. It was necessary to know what was being done at the institution and achieve an understanding of what is known about OER. Conducting these interviews also helped in discovering potential users for the system.

After the interview phase was finished a focus group with fellow digital library researchers and developers was organized to draw up an initial design of the system. Focus groups are a somewhat informal technique that can help in assessing user needs and feeling both before interface design and long after implementation [Nielsen 1997]. The reason behind using this method to create the system prototype was to collect ideas and further knowledge on how to design the solution for this project's goals.

These two stages are discussed in the rest of this section.

### 3.1.1 Interviews

The interviews were conducted before the launch of OpenUCT[1] and UCT's signing of the Berlin Declaration[2] in 2010 and 2011 respectively. In total, 6 people were interviewed: four professionals in education with technology; an OER coordinator at one of the faculties; and a professor experienced with OER at a high level.

The information collected from all the interviews is collectively stated below, categorized into facts, opinions and challenges:

**Facts**

- There are three ongoing projects at UCT that are differently related to this research. All the projects are independent and not collaborating.
- *pubs.cs*[3] is a Computer Science e-prints departmental repository. All publications produced at the department are ingested into the repository and posted

[1] http://opencontent.uct.ac.za/

[2] http://blogs.uct.ac.za/blog/oer-uct/2011/11/02/university-of-cape-town-signs-the-berlin-declaration-on-open-access-to-scientific-knowledge

[3] http://pubs.cs.uct.ac.za/

on the Web user interface. Access to the contents is free and open. Currently, the repository is accessed from around the world and has been proven successful [Suleman 2006]. A similar repository is LawSpace[4] at the Law faculty.

- *Law OER* is a potential project of the Law faculty at UCT that plans to provide coursework to students via an open repository with a content management system.
- *UCT OpenContent*[5] is a current project recently launched by the Centre for Educational Technology (CET). This project aims to build an index of potentially open resources from around UCT.
- At the current time, there is no intention to build one open repository for the whole University.

**Opinions**

- Integration of the proposed system with VULA[6], the University's learning management system, may be useful to lecturers or course conveners. It seems like a tiresome process to ingest the same materials more than once into different repositories or websites.
- Raising awareness of OER, its potential and importance among educators, University staff, students and across other institutions is a key aim.
- Simpler publishing or deposit tools are needed and will be sufficient for future enhancements when working with OER.
- The search facility in a repository can be crucial for access and sometimes get users frustrated when they fail to find specific contents.

**Challenges**

- There is a lot of interest in OER but not much knowledge from the educators. The CET is continuously trying to raise awareness of OER through presentations, projects and workshops.
- The biggest challenge faced by personnel and educators at UCT is the question of having the rights to publish material. The reason behind this is that educators are not expected or obliged to create their own educational materials by the University. Hence, course materials and resources are created by other materials they have access to.
- At the Department of Health Sciences, the OER initiative is concentrating on the process of creating OER instead of where these materials will be stored.

[4] http://lawspace2.lib.uct.ac.za/dspace/
[5] http://opencontent.uct.ac.za/
[6] https://vula.uct.ac.za/portal/

Some obstacles faced with using technology are slow Internet connections, courses are too wide, and patient privacy is a problem when trying to share videos and photos. Open licensing is also an obstacle.

- Motivating authors to contribute is difficult because there is no direct reward for sharing and the process is tedious.

**Summary of interview results** There was a clear knowledge of OER but awareness across all was an ongoing mission. Some OER initiatives and projects were in motion. It was clear from the interviewee comments that simpler solutions need to be introduced to assist and motivate educators and content creators in OER publication.

### 3.1.2 Prototype design

A focus group was set up to design an initial prototype for the system. The group was composed of 5 postgraduate students from the Digital Libraries Laboratory and the main researcher. The group was introduced to the project idea then asked about their perceptions and opinions on the application design. A paper prototype was drawn and general features were discussed. The group discussion lasted about 74 minutes. Figure 3.1 shows the prototype and recommendations from the focus group.

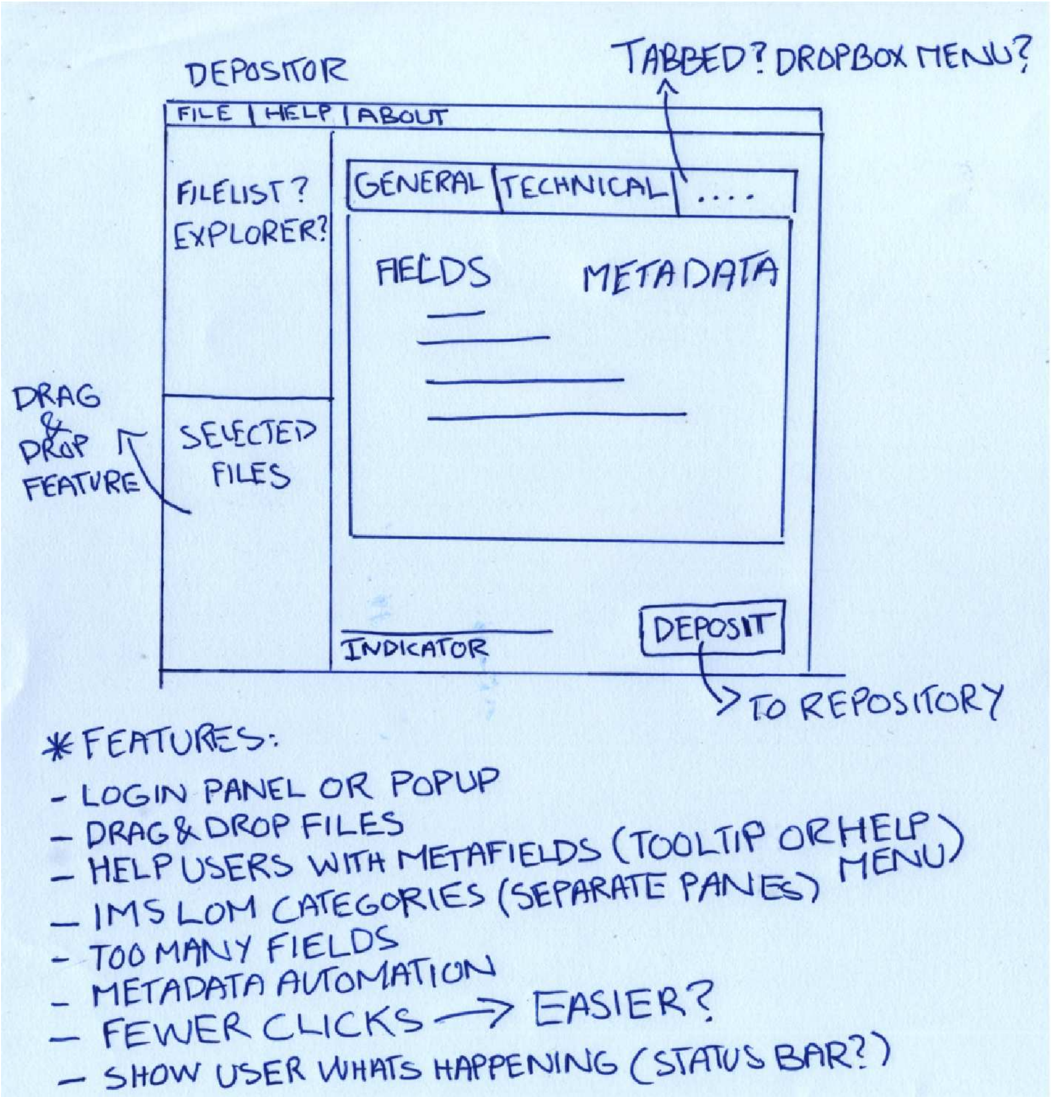



Figure 3.1: Initial system prototype

#### 3.1.2.1 Aims

The finalized aims of the desktop application are:

- deposit resources to repositories using deposit protocol
- provide user with required and optional metadata entry fields
- read resource files to automate technical metadata extraction
- present user with a status bar to indicate what the application is doing
- allow user to select resources

- allow the user to drag and drop resources
- provide help for user when entering metadata
- allow user to provide credentials for target repository

#### 3.1.2.2 General process description

The typical user deposit scenario was designed as shown in the interactivity diagram in Figure 3.2.

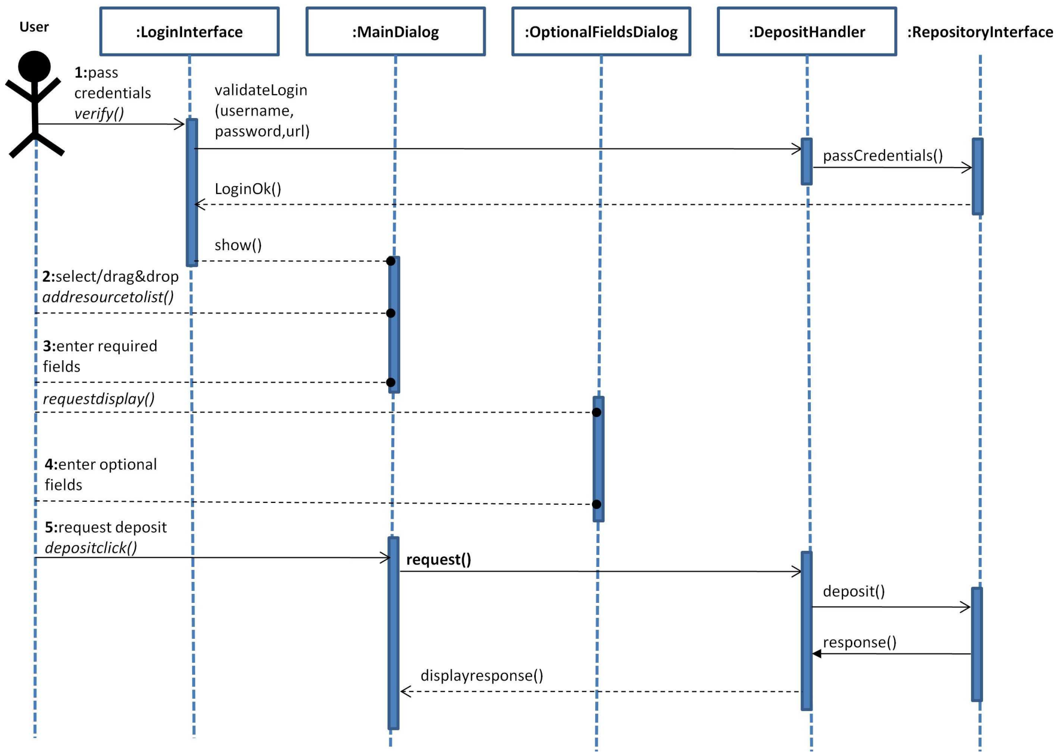



Figure 3.2: Application interactivity sequence diagram

A typical user interactivity sequence is as follows:

1. A user logs into the desktop application using credentials for a repository he/she is willing to deposit to.
2. Credentials are checked and the user is presented with a Main panel where he/she can select or drag&drop the resources.
3. On the same panel the user fills in the required fields.
4. The user clicks on a optional fields button to fill in some optional metadata.

5. The user clicks a deposit button. The deposit request is checked by a deposit handler and the deposit request is forwarded to the aformentioned repository. The repository responds with a success or failure message that is displayed to the user on the Main panel.

## 3.2 System Overview

The system comprises a repository, a desktop application and the SWORD client/server components. The desktop application was named ORchiD. ORchiD is an acronym for Open educational Resources Depositor. The main function is to package a resource with its metadata and send it to the repository at the user's request. The system makes use of a number of technologies and existing applications. Figure 3.3 is an overview of the system, showing the main components.

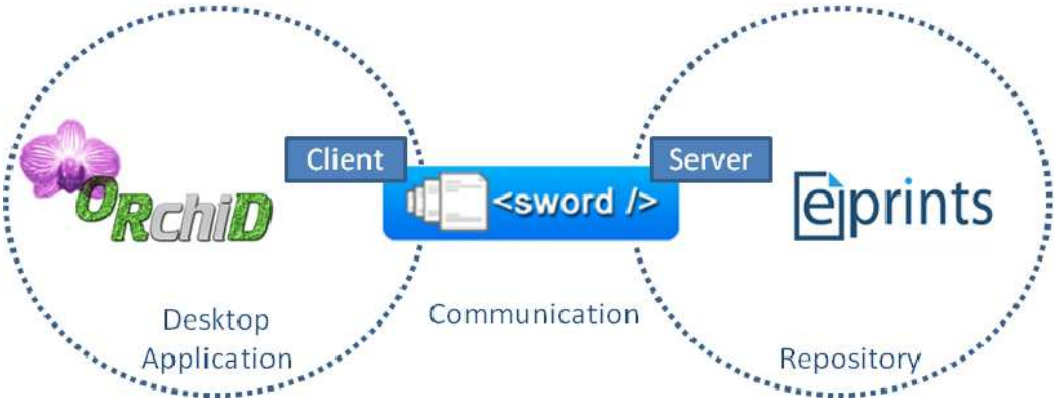



Figure 3.3: System Overview showing ORchiD, the repository, and SWORD

The following sections discuss these components and the main system functions further.

## 3.3 Technical environments

As suggested, existing technologies were integrated to develop the desktop application. Building and adapting the repository was also done with the use of several technologies. These technologies are listed below separately as they are two stand-alone solutions. The communication between the two systems is the SWORD protocol. Namely, the SWORD client was part of ORchiD and the SWORD server was a plugin already introduced in the EPrints3 repository system.

**ORchiD's** technical environment used:

- Eclipse Java EE IDE for Web Developers (Indigo Release) to develop the desktop application.

- Java(TM) SE Runtime Environment (build 1.6.0 26 03) to execute and run.
- SWORD v1.3 Java Library to develop the SWORD client
- GIMP v2.6.11 to create the icon and logo. The original photo of the orchid[7] (a photo of a Dendrobium Kingianum) is attributed to Anne Stauf.
- JTattoo v1.3[8] public release for the Look'n'Feel of the Java Swing components used.

ORchiD was eveloped on a machine with the following specifications: Ubuntu Linux 11.10 operating system; Intel Pentium Dual at 2.20Ghz;80GB storage hard drive; and 2GB memory.

**Repository's** technical environment used:

- EPrints Digital Repository Software version 3.2.8 (Apple Crumble) to build the test repository system.
- Apache Server v2.2.20
- Perl v5.12.4
- MySQL Distribution 5.1.58 for the underlying storage layer for the repository.
- Perl to edit EPrints files, plugins and modules.
- The EPrints server machine with the following specifications: Ubuntu Linux 11.10 operating system; Intel Pentium Dual at 2.20Ghz;80GB storage hard drive; and 2GB memory.

## 3.4 The Repository

In this system an EPrints repository was installed and configured to work as a test repository[9] for the desktop application. The first change made to the repository was increasing files supported by the repository system. The reason behind this is that when the repository was unable to identify the file it could not represent the learning object correctly and treated the file as a data stream and stored it without any handling.

The metadata representation of the objects and the deposit protocol are the significant changes made to the repository system. Firstly, the IMS Learning Resource Metadata was the standard chosen to represent the objects and hence the repository was adjusted to conform with the IMS specifications. Secondly, an import plugin was needed to translate the metadata deposited into EPrints XML files and ingest the metadata into the repository. Thirdly, the repository needs to support remote deposit of resources to allow the desktop application to deposit the user's resources and this is enabled by the support of the SWORD plugin at the server side.

[7] http://www.flickr.com/photos/annestauf/5172750532/
[8] http://www.jtattoo.net/
[9] http://mnour.cs.uct.ac.za/

### 3.4.1 Additional file support

Adding extra file support to the repository was not staightforward due to the number of files needed to be edited to add handling of one extra file format or *MIME type*. A *MIME type* is an Internet media type that is a standard identifier of files. This standard states that an identifier consists of at least two parts like `text/html` and `video/mp4` for Hypertext Markup Language and MPEG-4 videos respectively.

The process of adding support and handling of file types to EPrints is clarified by the use of the following example. To add support for 3rd Generation Partnership Project (3GPP) video with MIME type `video/3gpp` the following steps were taken.

1. add the MIME type to the list of recognized MIME types by the repository in the `document` file in the namedsets folder inside the archive directory

```
audio/3gpp
```

2. map file extensions of the MIME type in the Perl configuration file *document_upload.pl*

```
$c->{mimemap}->{3gpp} =  "audio/3gpp ";
```

3. associate a phrase for the new file type that is used when a resource of that type is displayed on the repository website

```
<epp:phrase id=<"document_typename_video/3gpp">3GPP video</epp:phrase>
```

4. associate a file icon to appear in the representation of this file on the repository website

### 3.4.2 Metadata Representation

For the repository to comply with the IMS Metadata standard, three of EPrints' configuration files were edited: the *eprint_fields.pl*, file which contains all the possible metadata fields an eprint can have; the *default.xml* deposit workflow file, which defines the input fields for each eprint type; and the *eprint_fields.xml* file, which specifies the phrases for the names and helptext of the fields. Since the eprint type in the system is any educational resource, the workflow would be the same for all objects. The configuration files are found in the *cfg* folder in the repository's directory. A metadata element is referred to as an eprint field in EPrints. The aim is to have all the IMS Metadata elements represented as eprint fields so that they could be handled by the repository during transfer without risking the loss of any metadata instances. As an example, consider the data element `Structure` from the `General` category of a LRM object. This element has a controlled vocabulary. The steps to add the field are shown next.

1. Defining the type and attributes of the field in the *eprint_fields.pl* file:

```
{
        'name' => 'structure',
        'type'  =>  'set',
        'options' => [
                            'atomic',
                            'collection',
                            'networked',
                            'hierarchical',
                            'linear',
                        ],

        'input_rows' => 1,
},
```

2. Adding the field to the *default.xml* deposit workflow file:

```
<component><field ref="structure" required="no" /></component>
```

3. Adding the display name(s) and the help text to *eprint_fields.xml* phrases file:

```
<epp:phrase id="eprint_fieldname_structure">Structure</epp:phrase>
<epp:phrase id="eprint_fieldhelp_structure">Please select the organisational
structure of this learning object</epp:phrase>
<epp:phrase id="eprint_fieldopt_structure_atomic">Atomic</epp:phrase>
<epp:phrase id="eprint_fieldopt_structure_collection">Collection</epp:phrase>
<epp:phrase id="eprint_fieldopt_structure_networked">Networked</epp:phrase>
<epp:phrase id="eprint_fieldopt_structure_hierarchical">Hierarchical</epp:phrase>
<epp:phrase id="eprint_fieldopt_structure_linear">Linear</epp:phrase>
```

Finally, the database tables are regenerated and EPrints is restarted so the changes can take effect. The repository caters for all the IMS LRM elements but also allow a subset representation of a resource by an instance.

At this stage a special case was identified. EPrints repository system is NOT capable of handling or accomodating compound fields inside compound fields. In the metadata implemented for the test repository this case was present two times: the

`Taxon Path` compound field in the `Classification` category, which has a compound subfield `Taxon` with `Id` and `Entry` as subfields; and the `Resource` compound field in the `Relation` category, which has a compound subfield `Identifier` with `Catalog` and `Entry` as subfields. A simple solution was used to represent these fields. The fields were represented as an XML chunk to preserve the values and prevent any loss of metadata.

Figure 3.4 shows all the fields added to the repository. EPrints provides a variety of field types, attributes and options to represent fields[10]. For instance, field *types* like `set` and `compound` are used for vocabularies and fields with subfields respectively.

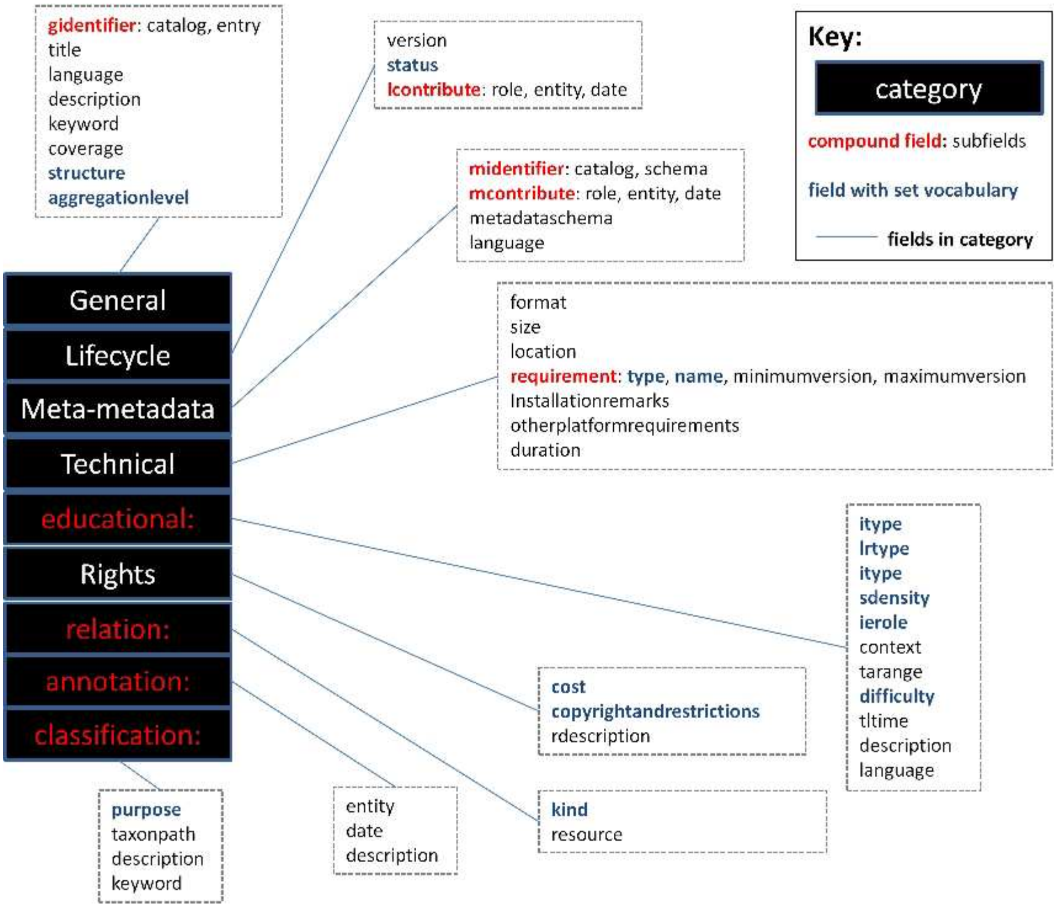



Figure 3.4: Metadata fields added to the EPrints test repository

### 3.4.3 IMS import plugin

A default installation of EPrints 3.2.8 includes a set of import plug-ins. One of those is the IMS import plug-in. The plug-in only ingests the `title` and `description` IMS fields, mapping them to the EPrints metadata fields of title and

[10] http://wiki.eprints.org/w/Category:EPrints_Metadata_Fields

abstract respectively. It also stores the XML file, avoiding loss of metadata but without representing the rest of the fields. For this reason, the IMS import plugin, *IMS.pm* was re-written to read the rest of the metadata fields and map them to the created fields explained in section3.4.2.

The IMS import plug-in takes a package sent by a client as input. It unpacks the contents and finds the XML file containing the IMS metadata. The file is opened and the field values are extracted. The values are mapped directly to the corresponding fields in the repository. The resource files are imported into the repository with the read metadata.

At this stage, it is important to mention how mapping of the IMS metadata fields occurs and what type of additions were made to the plug-in. A `PERL` XML DOM parser was used and the field mappings are presented, showing how an import is achieved. The main difference is whether a field is single or compound. The code snippets are shown next for clarification.

**SINGLE** holds one or several values for one tag. The snippet shows the lines to try to get the value in `<tagname>`, place it in a `$fieldname` and add the value of the string to `$epdata`. `$epdata` is the string that holds all the mappings from an instance of an IMS metadata file. When there is no value the parser jumps to the next line of code. If this field was specified multiple times then an array is used instead of a string and all values are pushed into the array.

```
my $fieldname = ($tag_wrap->getElementsByTagName( “tagname" ))[0];
        if ( defined $fieldname )
            {
                $epdata->{epfieldname} = EPrints::XML::to_string(
                EPrints::XML::contents_of( $fieldname ) );
            }
```

**COMPOUND** holds one or several values for more than one tag. The snippet shows how the parser deals with a multiple compound field. The multiple compound field `<tagname>` has subfields `<subtagname1>` and `<subtagname2>`. The parser finds all `<tagname>` entries and places then in the array `@cpdfield`. The parser iterates through each `<tagname>` and takes the values in each subfield. The values with their associated fields are placed in the array and the array is pushed to `$epdata`. `$epdata` is the string that holds all the mappings for an instance of an IMS metadata file.

```
my @cpdfield_wrap = ($tag_wrap->getElementsByTagName( “tagname” ));
    if(defined $cpdfield_wrap[0])
        {
         my @cpdfield;
         my $cpdfieldNode;

        foreach $cpdfieldNode( @cpdfield_wrap )
            {
                my $subfield1 = ($cpdfieldNode->getElementsByTagName(
                “subtagname1” )-> item(0)->getFirstChild->getData);

                my $subfield2 = ($cpdfieldNode->getElementsByTagName(
                “subtagname2”)-> item(0)->getFirstChild->getData);

        push @cpdfield, {epsubfield1 => $subfield1, epsubfield2 => $subfield2};

            }
             $epdata->{epcpdfield} = [@cpdfield];
        }
```

### 3.4.4 SWORD deposit interface

SWORD was introduced as a plug-in on a default installation of EPrints 3 repository software. The deposit interface can be activated and deactivated by changing the SWORD configuration file. The server supports the use of HTTP Basic Authentication and HTTP Post to provide the client with the ability to interact with the repository. The client is able to authenticate a user, retrieve a service document and deposit a supported package that contains a resource with its metadata.

The interface location for all SWORD interaction with the test repository is:

```
http://mnour.cs.uct.ac.za/sword-app/
```

#### 3.4.4.1 Configuration

An EPrints SWORD configuration file defines how the interface should work. Definitions of type of files, user mediations, deposit collections and supported packages are configured in this file. These are then presented in a Service Document that tells deposit clients on a request what the repository has to offer through the SWORD interface.

In the test repository, the plugin is configured to accept all types of files in IMS Content packages. The location of a deposit is pointed directly to the live archive.

#### 3.4.4.2 Testing

This interface was tested using example clients produced by the SWORD project. Sample IMS content packages were successfully deposited using command line and Web clients. The following points state how a test is executed.

1. The client requests the Service Document from the repository with user credentials.

2. The server authenticates the user and responds with a Sevice Document.

3. The client reads the Service Document and retrieves the deposit location.

4. An IMS package is sent to a specified collection in the repository by the client.

5. The SWORD server performs a deposit and responds with a success or failure to the client.

After the test repository was ready to accept SWORD requests, store all of the expected metadata and present OER objects as designed, the development of the desktop application was resumed.

## 3.5 The Desktop Application - ORchiD

The desktop application consists of a number of components (See Figure 3.5) that work together to provide the user with the simple repository deposit. The development of the application exploits some of the Java technologies including the Drag and Drop data transfer features and Swing GUI widget toolkit. The application also takes advantage of the SWORD Java library to develop the client that interacts with the server and sends requests to the repository.

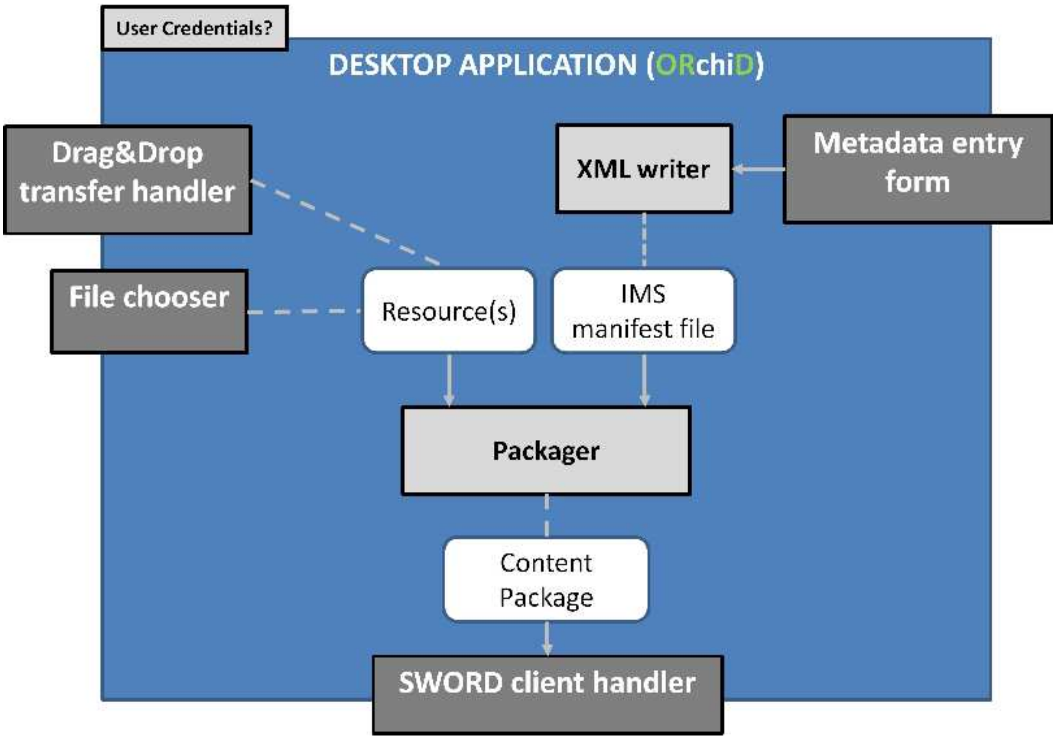



Figure 3.5: Main components of the desktop application

### 3.5.1 Resource handling

The desktop application window contains a list component where the resource files the user chose are listed. The user is presented with two ways to add a file to the list (See Figure 3.6). One is to drag the resource from the desktop and drop it onto the list component. Java Swing's Drag and Drop is used for its implementation. Another way is by clicking on a button that displays a regular file chooser in which a file can be selected and added to the list.

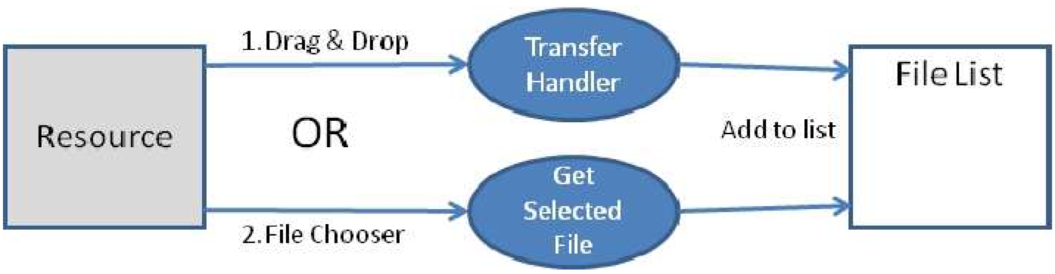



Figure 3.6: Adding resources to list

The `File List` shown in Figure 3.6 is a Java List component that stores the locations of the files placed in it. Placement of the physical file locations is done using two handlers: a TransferHandler for the Drag&Drop feature; and a Java FileChooser for the Browse&Select feature. The TransferHandler reads a Drag action as a Gesture, and once the mouse button is released over the File List, it trys to read the file location from the object released. Nothing happens if anything but a file is released in to the File List. The other way is using the FileChooser that works like any other traditional file chooser in which it retrieves the selected file location at a button click.

### 3.5.2 Metadata entry

The Learning Object Metadata data model specifies a hierarchical structured set of elements. A LOM instance consists of nine categories and each category consists of a number of data elements that describe a learning resource. The desktop application provides the user with a form allowing the entry of element values. The elements have different data types, some have sub-elements and some have controlled vocabularies which the entry form accommodates.

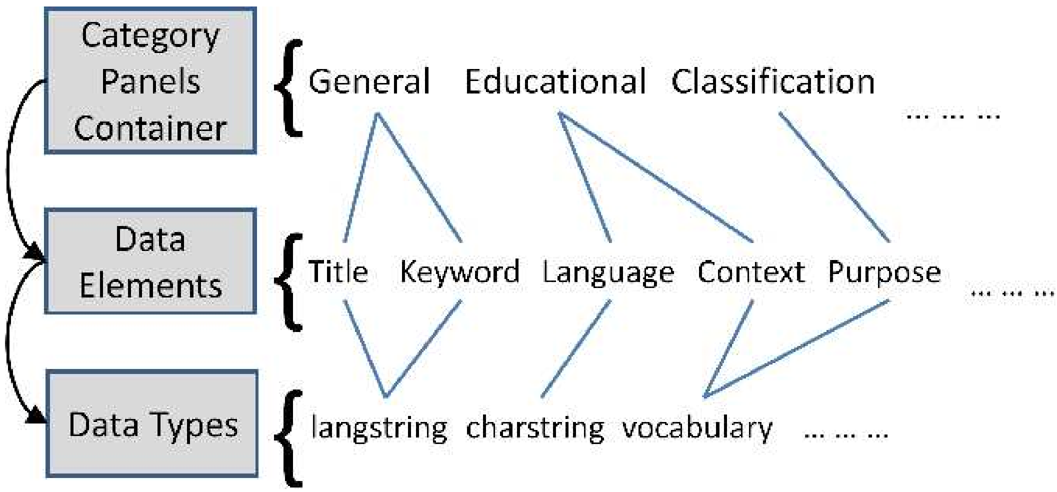



Figure 3.7: structural view of the IMS Metadata implementation

Figure 3.7 shows how the metadata structure was implemented. Together they form a `lomObject` and each part is considered as a user interface `lomComponent`. Each `lomComponent` translates to an area in the end user interface. The three main parts shown in the figure are described as follows.

**Category Panels Container** includes the nine categories specified in the IMS LRM. Each of the categories contains a set of data elements and is translated as a window panel in the user interface.

**Data Elements** each correspond to and represent a field in the metadata specification. These also hold attributes that show if a field is `Single`, `Multiple` or

`Compound`.

**Data Types** correspond to a value in the IMS manifest file. It determines the expected and controlled value entered by the user. Each Data Element is given a certain Data Type that conforms with the recommended types by the IMS specificaions.

Due to the complexity of the IMS specification, the application offers an option for the user to enter values for fewer elements. Other than the basic user credentials, the `Title` and `Description` fields are the only required fields in the data set used in this application.

### 3.5.3 IMS XML Manifest writer

The main purpose of this component is to create the *imsmanifest.xml* file after a user had selected a resource, entered the metadata and clicked the `Deposit` button to initiate the deposit action. The writer takes the entered values, creates an XML document and writes the values entered by the user to their appropriate places. This component uses the Java I/O library to create the manifest. A `BufferedWriter` is used to write the sections of the manifest file and a `FileWriter` to create the XML file on disk.

An XML file is constructed once a deposit is initiated. A partially completed manifest tree is written with the standard attributes and tags. The file declaration, the manifest plus LOM attributes and regular structure of any manifest file, as described in 2.3.2, is written first. Then the corresponding tags to the filled-in fields are added with their values. Finally, the list of files added by the user are referenced in the `<organization>` and `<resources>` tags. The XML manifest file *imsmanifest.xml* is closed and saved in a folder with a copy of the files selected to deposit by the user.

### 3.5.4 Packager

The application complies with the IMS Content Packaging specification with this component. The specification states that a package, in a standard Package Interchange format, includes an IMS manifest file and resource(s). This component packages the metadata file created by the XML writer and the resource selected by the user into a zip file. The package is forwarded to the SWORD client to handle the posting. This is shown in Figure 3.8.

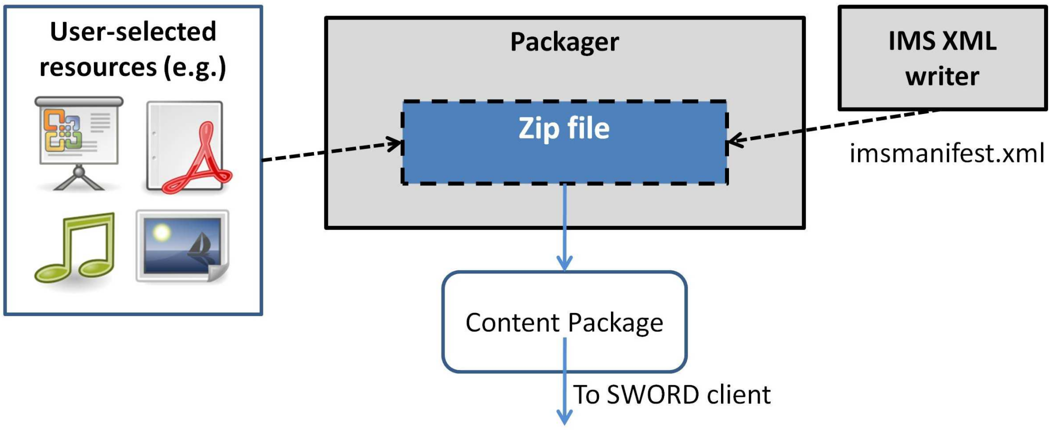



Figure 3.8: Content Package creation

### 3.5.5 SWORD client

The client submits a package to the SWORD server as a bit stream using an HTTP POST request. The post includes a Header that contains user credentials, type of the package and the bitstream that is the package itself. When the package is received the server sends a response back to the client about the success or failure of the deposit.

The SWORD project produced a Java SWORD library and a set of example clients that the application exploits for this component. The client should be able to query the repository for the service and request a deposit at the user's request.

The SWORD client main functions are implemented as follows:

**isLoginOk** checks the user credentials and authentication by retrieving a `ServiceDocument` from a specified `URL`.

**getServiceDocument** attempts to retrieve the service document from a given `URL` using a specified `Username` and `Password`.

**getDepositLocation** attempts to read a `ServiceDocument`, find the deposit location and return its `URL`.

**depositFile** attempts to deposit a package with a given `Filepath` to the deposit `URL` using the given `Username` and Password.

### 3.5.6 The user desktop interface

The user desktop interface allows the user to perform the actions provided by ORchiD and make use of the SWORD client. The main aim is to create a content package and post it to a SWORD-compliant repository, in this case the test repository discussed in Section 3.4. The interface attempts to complete the deposit in a simple manner and with as few user interactions as possible without losing the richness of the metadata.

#### 3.5.6.1 Login

The user is presented with a Login window when ORchiD is first executed. Figure 3.9 shows the login window. The user is presented with spaces to fill in credentials and a Repository Service Document URL. A Service Document URL is provided by any SWORD-compliant repository. The default address is that of the test repository described in Section 3.4.

When a user clicks on the *Login* button, the credentials are checked by the retrieval of the Service Document. Typically, a successful retrieval means the user has the proper credentials. The SWORD client performs this check using the `isLoginOK` function described in section 3.5.5. At a successful authentication the user information is stored in a file so the next time ORchiD is executed the user is automatically logged in. A *LogOut* menu option is also provided.

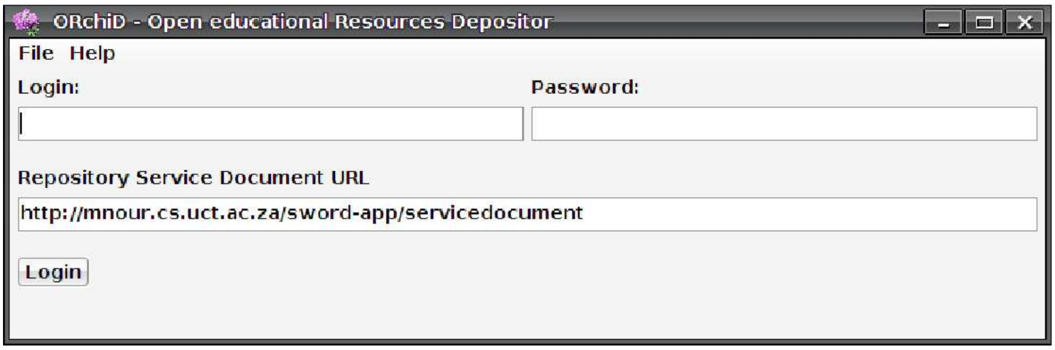



Figure 3.9: Login panel displayed when ORchiD is executed for the first time

#### 3.5.6.2 Main Panel

After a user is authenticated the window shown in Figure 3.10 is displayed. The user constructs the content for the resource at this stage. The user is required to enter a `Title` and `Description`. The large white box represents the *Filelist* where the user adds resources. The window also shows an Upload button to initiate the deposit process and a Reset button to clear all the fields. An `Optional Details` tab, which contains the rest of the metadata fields, is also accessible from this window.

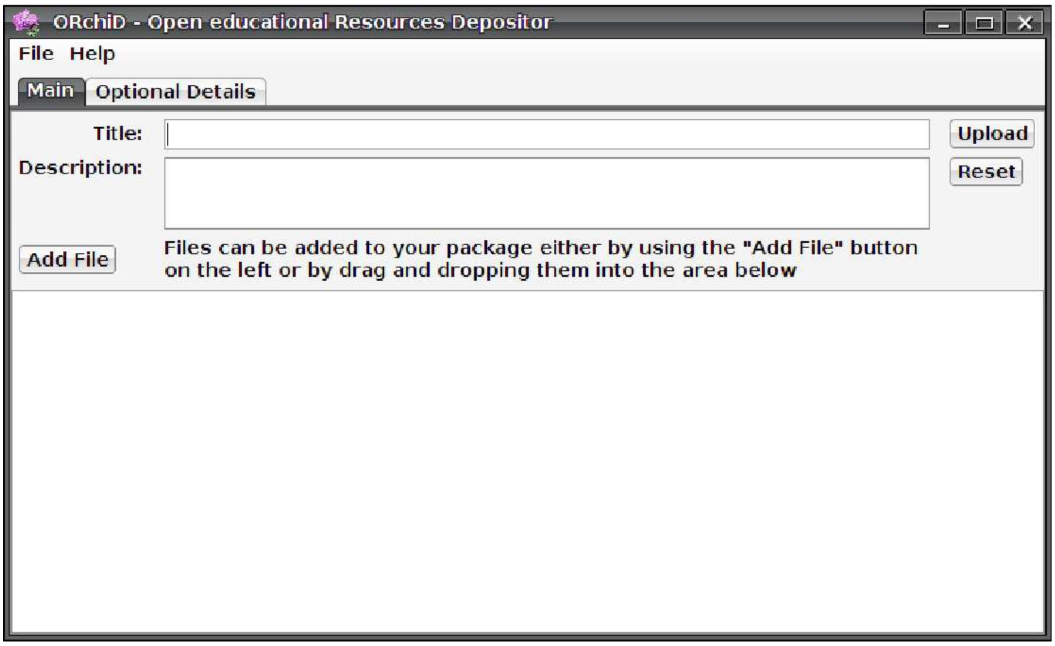



Figure 3.10: Main Panel displayed after a successful Login

#### 3.5.6.3 Optional Details panel

When a user clicks on the `Optional Details` tab on the Main Panel the rest of the metadata fields are presented. The window in Figure 3.11 shows how the rest of the fields are presented. The user is able to navigate though the categories using the combobox on the top right corner. When a certain category is selected, the inner frame is drawn to represent the fields that category accomodates. The user fills the fields as needed and returns to the main panel to initiate the deposit.

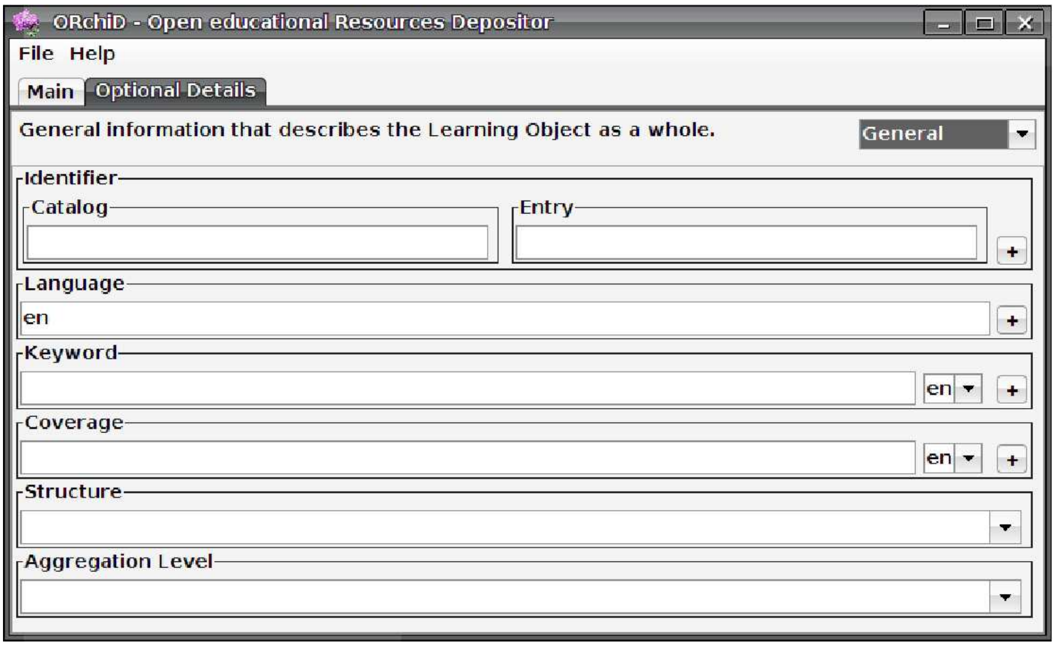



Figure 3.11: Optional Details panel for metadata entry

## 3.6 Summary

A desktop application, ORchiD, was developed to simplify depositing into repositories. It comprises a front-end interface for user input, an XML writer that transforms the user-entered fields into an XML file, a packager that compresses the selected resources with the created metadata file into a package and the SWORD client that handles the transfer. A repository was also configured to be a test destination for the solution. In the next chapter an evaluation of ORchiD is presented.

Chapter 4

# Evaluation

## Contents



ORchiD is built to help educators or resource creators to share their educational resources online. The tool presents the user with a way to deposit into SWORD-compliant repositories directly from the desktop. The main goal of the evaluation is to prove that there can be an easier way of creating OER with comprehensive and descriptive metadata. This chapter presents the evaluation of three aspects of the project: the repository integration, where the changes and compromises made such that the usage of the repository is seamless to the user are discussed and presented; the metadata representation, where the extent of the descriptiveness and usefulness of the metadata fields is interpreted; and usability testing, where actual content creators interact with the repository using the desktop application and provide their feedback.

## 4.1 The repository integration

An attempt to adapt the respository structure was taken in this project. Several issues were faced when integrating the test repository with ORchiD. Some of these were resolved and others needed a certain workaround. Table 4.1 lists these issues and states how they were handled.

| **Issue** | **Description** | **Resolved** | **Workaround** |
|---|---|---|---|
| *1* | handling a limited set of file formats | ✓ | |
| *2* | inability to handle compound in compound fields | | ✓ |
| *3* | sub-fields inherit multiplicity attribute from parent field | | ✓ |
| *4* | IMS plug-in parsed only Title Title and Description fields | ✓ | |

Table 4.1: Issues with the repository integration

The handled issues were described in Chapter 3. The implementation was able to overcome all the issues raised by the design. It may seem like a considerable number of issues but, nevertheless, the repository was successfully integrated with ORchiD to accommodate OER.

## 4.2 The metadata representation

| **IMS LOM Specification detail** | **ORchiD** | **EPrints test Repository** |
|---|---|---|
| Langstring | textboxes with language attribute | English language set by default but no special specifics for language handling |
| DateTime | set of boxes for entering date and time | represented as date |
| controlled vocabulary | drop down lists | set values and options |
| compound field | grouped together in panel and specified attributes for each | represented as a table of values and subfields inherent attributes from parent field |
| compound field with subfield as compound field | a panel in a panel | Not possible. Inner compound fields were presented as XML |

Table 4.2: IMS Learning Object Metadata presentation in ORchiD and the EPrints repository

The metadata representation was technically correct for all the fields by all the components except for the "compound in compound fields" in the repository website. This detail or property represents three metadata fields. The workaround was to display them as human-readable XML tags in the EPrints test repository. This could be resolved by the use of automatic renders at the repository's side but this was out of the scope of this project and would be considered as future work.

## 4.3 User Study

To test the effectiveness, efficiency and usability of ORchiD, a user study was conducted. Users were asked to perform some tasks and fill in a questionnaire. This section presents the survey design and shows how the survey was tested pre-launch through a pilot study. Appendix A shows the survey and the questions the users were presented with.

### 4.3.1 Population Description

The typical provisioned user for this application was anyone who owns a digital object that has any potential learning impact and is willing to share it publicly online. This property describes a vast and dispersed population. The sample population was educational content creators in universities and industries around the world. The main reason behind this selection is that it provided a sense of easy accessibility and cost effectiveness. The focus of the project is OER and hence, the assumption that anyone from this group has or produces some kind of learning object on a regular day was made. There were no special restrictions but user control was done through email to keep track and count of the participants. An invitation (See Appendix A.1) was sent to 178 individuals via email. The user specifications ranged from professors to teachers in many educational institutions. Some individuals or educational professionals from industry were also contacted.

### 4.3.2 Survey design

The survey was designed in such a way to deal with the accessability and time constraints regarding the target population. LimeSurvey[1], an open source online survey tool, was used to create the survey (see Appendix A.2). The only contact method between the researcher and participants was through email. The problem with this type of remote testing is observational data could not be taken and not considered in analysis. The advantage was placing the participants in real life scenarios. Two basic scenarios were considered for this survey: a lecturer sharing class slides as an OER at the end of a lecture and a course convenor willing to share a set of course resources at the end of a term.

The survey contained both structured and unstructured [Murray 1999] questions. Structured questions are are quantitative questions used when the participant is pre-

[1] http://www.limesurvey.org/

sented with a fixed set of choices and no new ideas are needed from the respondent. These simplify data collection and analysis and need less time to answer, e.g., *"Do you have a running copy of ORchiD? Yes or No"*. Rating questions are also considered as structured questions. A Likert scale [Likert 1932] from *Strongly Disagree* to *Strongly Agree* was used for rating in this survey to evaluate the extent of which the participant agrees with statements about the different aspects of the system (see Appendix A.2). Unstructured questions are qualitative open ended questions were the participants thoughts and ideas are needed. These were used here to give the users a chance to elaborate and express other thoughts they were not asked about specifically in the rest of the survey.

Figure 4.1 shows the flow of this survey.

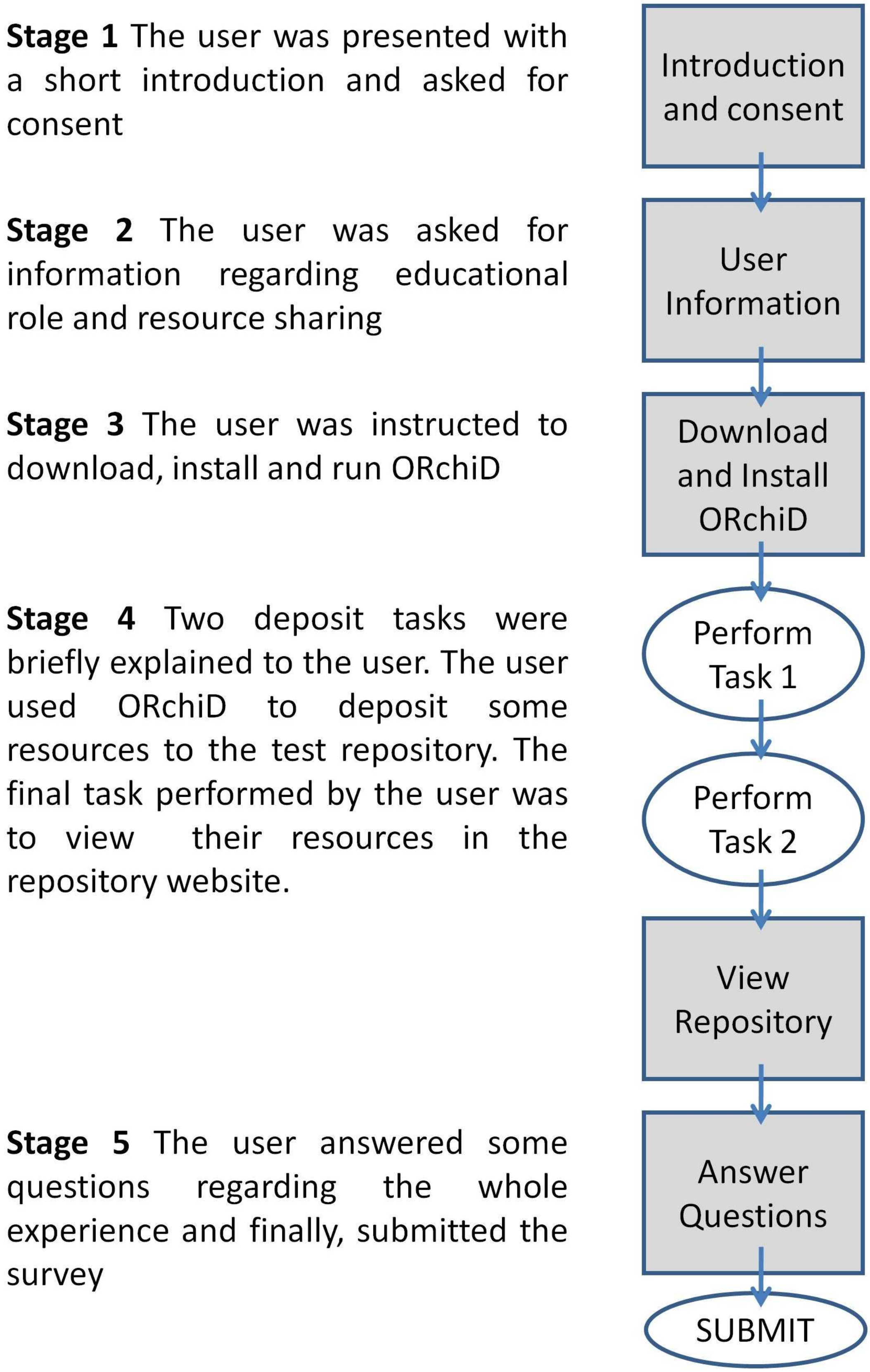



Figure 4.1: Survey stages diagram

### 4.3.3 Pilot Study

Before commencing the survey a pilot study was conducted. The purpose of the pilot study was to identify any complications with the structure of the survey and any critical fixes needed to be made to ORchiD. Six postgraduate students were involved in this study. There was no one-on-one observation involved in this study to mimic the real survey scenario but post-study discussions were held to clarify some of the issues raised. The survey post was sent to the students and they were informed of the purpose of the pilot study.

Table 4.3 lists the issues raised and summarizes how they were resolved before the actual survey was started.

| **Issue identified** | **Resolution** |
|---|---|
| ***related to Survey*** | |
| Inconsistencies of file names between survey and the physical files | File names were changed and references were corrected |
| Tooltips and drag & drop features were not realized immediately | Tips were added to the survey at the stage of presenting the tasks |
| Typographical errors and use of too technical terms in survey | The identified errors were corrected and the terms indicated were changed |
| ***related to ORchiD*** | |
| There is no indication that the application is busy at the point of upload | The application was set to lock when uploading, the status bar messages where made clearer and a busy mouse icon was shown |
| The application did not timeout when there was no Internet connection | A timeout was set and a 'failed' message was displayed on the status bar |
| Resources appear 'Under Review' and not in the Live Archive | This was corrected and SWORD server was set to accept deposits to the Live Archive |

Table 4.3: Issues identified from survey pilot study

## 4.4 Results

In this section the survey responses are analysed. The total responses are indicated but only the completed responses were analysed.

### 4.4.1 Participants

As indicated in Section 4.3.1, 178 individuals were invited to participate through email, including 27 automatically generated replys indicating the email recipient was unavailable on email for differen reasons. There was 99 (%55) responses for this survey: 74 incomplete responses; and 25 fully completed responses. Considering only the completed responses, this survey had a %14 response rate. Interestingly enough, the EPrints test repository had 51 registered users at the end of this survey. After agreeing with the terms of the survey the participants were asked to select their educational user roles and choose from a list which systems they have used to publish or share their resources before (see Appendix A.2).

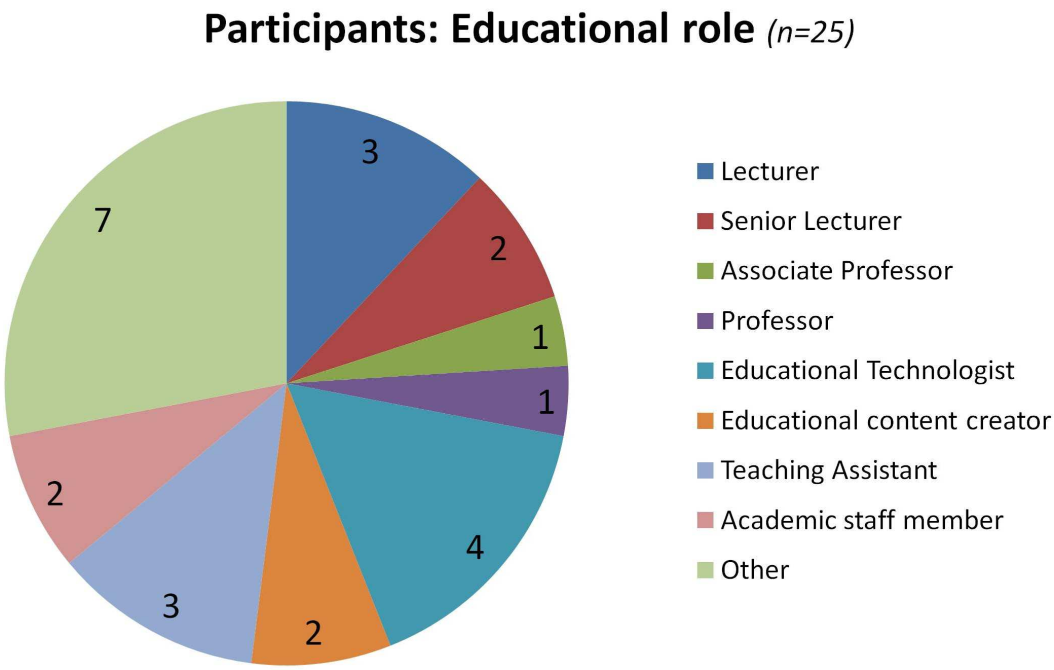



Figure 4.2: Participants educational roles chart

Figure 4.2 shows a pie chart indicating the diverse educational roles associated with the participants who took part in this survey. The 7 individuals who selected other indicated the roles listed below:

- OER project administrator
- Postgraduate student
- Executive of non-profit foundation
- Associate Researcher
- Recent graduate
- Former Professor
- Researcher

The users were asked to indicate other locations they deposited resources to. It was found that each of the participants deposited at least one resource to one of the listed systems. Figure 4.3 shows what the participants selected.

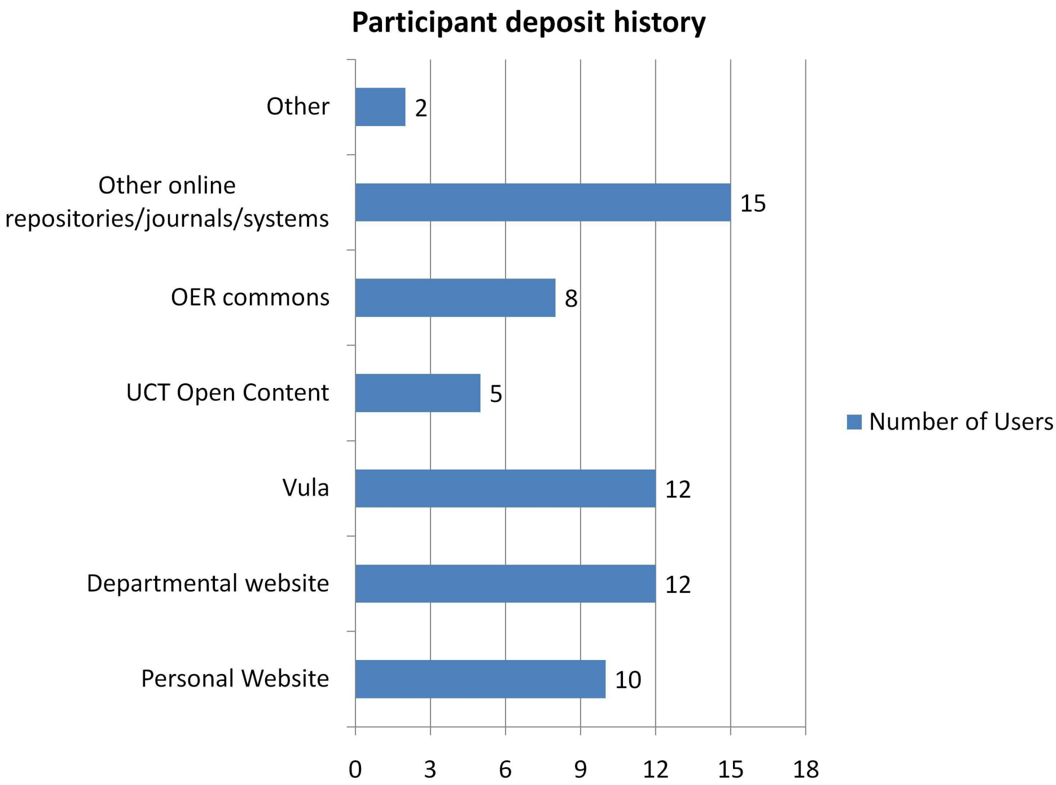



Figure 4.3: Previous deposit locations by users

### 4.4.2 Task Completion

The participants were asked to download and run ORchiD. They were then asked whether a running copy of ORchiD was available to them and were instructed on how to identify this (*See Appendix A.2*). The user was then presented with the tasks to perform and presented with questions for feedback on the task completion. Figure 4.4 indicates that not all users completed both tasks. It is also worth noting the two individuals that failed to complete the tasks were also not able to run ORchiD, as they indicated in their survey entry, but still submitted a survey response. It was hard to identify the reason why they were not able to run ORchiD because there was no contact between these two users and the researcher.

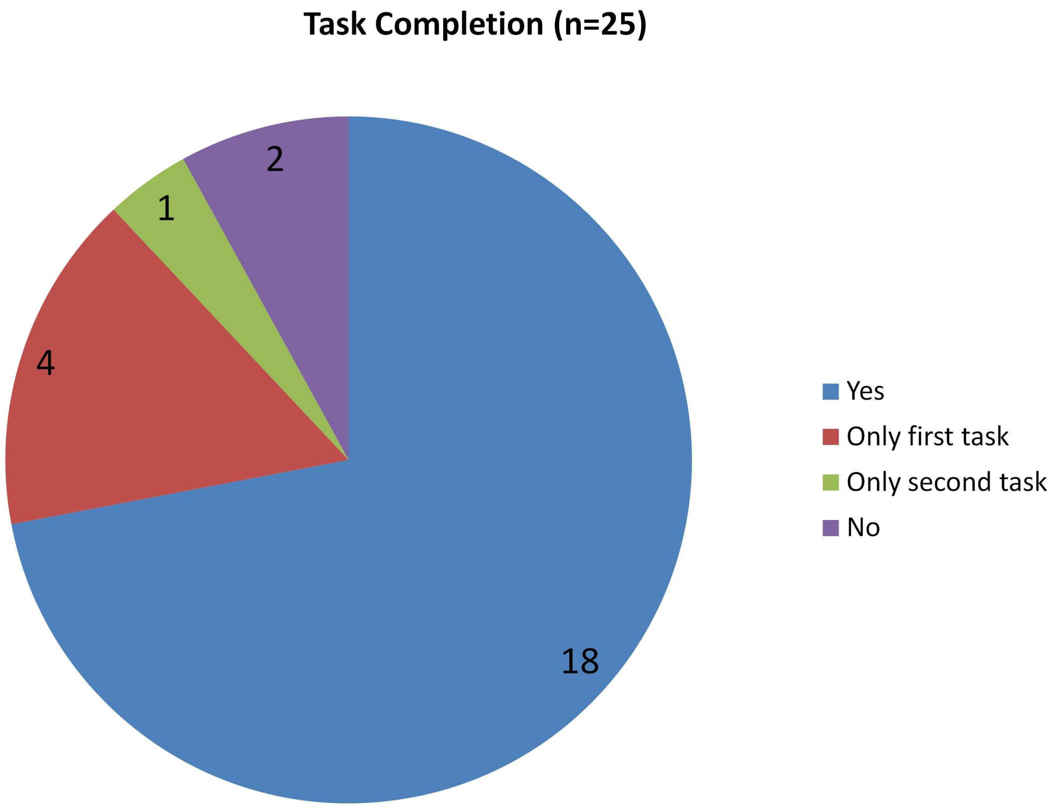



Figure 4.4: User task completion

### 4.4.3 ORchiD experience

The users were presented with a set of statements describing their experience when performing the tasks with ORchiD. A Likert-type question (see Appendix A.2) was used to determine the degree which the participants agreed with these statements. Table 4.4 lists the statements and the selection distribution from the participants. The **Mode** selection for each statement on the table is in bold numbers.

| **Statements about ORchiD** | No answer | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree | *MEAN* |
|---|---|---|---|---|---|---|---|
| I immediately understood what I needed to do | - | - | 3(12%) | 4(16%) | **10(40%)** | 8(32%) | 3.92 |
| The application was easy to work with | - | - | 4(16%) | 3(12%) | **11(44%)** | 7(28%) | 3.84 |
| I like the drag and drop feature | 3(12%) | - | - | 6(24%) | 5(20%) | **11(44%)** | 3.92 |
| I was able to describe my resource in the first task | 1(4%) | - | 2(8%) | 5(20%) | 7(28%) | **10(40%)** | 3.88 |
| I was able to describe my resource in the second task | 1(4%) | 1(4%) | 1(4%) | 4(16%) | 6(24%) | **12(48%)** | 3.96 |
| The optional fields were easy to comprehend | 1(4%) | 2(8%) | **8(28%)** | 6(24%) | 5(20%) | 3(12%) | 2.84 |
| The tooltip feature was useful | **7(28%)** | - | 3(12%) | 6(24%) | **7(28%)** | 2(8%) | 2.48 |
| In general, the application is simple | 1(4%) | 1(4%) | 2(8%) | 3(12%) | **12(48%)** | 6(24%) | 3.68 |

Table 4.4: User experience with ORchiD

As shown on Table 4.4 around 70% of the users either selected *Agree* or *Strongly Agree* for all statements about their experience. Except for the two statements *'The optional fields were easy to comprehend'* and *'The tooltip feature was useful'* only 32% and 36% selected *Agree* or *Strongly Agree* respectively. A possible reason for this is that the optional fields were not comprehensive or even familiar to the users. As for the tooltip feature, some of the users mentioned in their comments (See Section ) that it is an annoyance waiting for the tooltip to appear at each field and suggested a more comprehensive help feature or accompanying detailed document explaining all details of each field. ORchiD has succeeded to be understandable and easy as clearly shown by the rest of the statements.

### 4.4.4 Category representation and the Optional fields in ORchiD

The metadata entry form and the fields were described in statements. The user was asked to select an agreement level for each of the statements. Table 4.5 lists the statements and the number of user selections for each point in the scale. The average of selections is also listed. The **Mode** selection for each statement on the table is in bold numbers.

| **Statements about metadata standard used** | No answer | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree | *MEAN* |
|---|---|---|---|---|---|---|---|
| The fields are suitable ro represent my resources | - | - | 4(16%) | 7(28%) | **10(40%)** | 4(16%) | 3.56 |
| I expected other fields. Something is missing | 1(4%) | 2(8%) | **7(28%)** | **7(28%)** | 6(24%) | 2(8%) | 2.84 |
| More fields should be required not just the title and description. | 0 | 2(8%) | 4(16%) | 2(8%) | **13(52%)** | 4(16%) | 3.52 |
| In general, I am happy with the details presented for resource description. | 1(4%) | - | 2(8%) | 5(20%) | 7(28%) | **10(40%)** | 3.88 |

Table 4.5: User view of ORchiD metadata representation

Most users agreed that additional fields should be required for entry, not just the `title` and `description` fields. Nevertheless, as shown on Table 4.5, 68% of the users either *Agree* or *Strongly Agree* that they are happy with the resource description details presented by ORchiD.

### 4.4.5 Repository OER representation

The user was asked for feedback about the EPrints test repository representation of the ingested resources. The statements are listed in Table 4.6 with the user selections. The average selections are listed in the Mean column. The Mode is also represented as bold numbers.

| **Statements about resource presentation** | No answer | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree | *MEAN* |
|---|---|---|---|---|---|---|---|
| My items are represented correctly | 3(12%) | - | 1(4%) | 4(16%) | 7(28%) | **10(40%)** | 3.68 |
| The presentation of my items was as I expected | 2(8%) | - | 3(12%) | 5(20%) | 7(28%) | **8(32%)** | 3.28 |
| The information presented is easy to understand | 2(8%) | 1(4%) | 2(8%) | 5(20%) | 7(28%) | **8(32%)** | 3.52 |

Table 4.6: Repository representation of resources

In terms of how the user deposits were represented on the repository Website, Table 4.6 shows that at least 60% of the users agreed with all statements. According to the results, the repository representation was correct, as expected and understandable.

### 4.4.6 Comparison to Deposit history

The user was asked to give a comparison between previous deposit experiences and ORchiD in terms of difficulty. A scale of 1 to 5 was presented to the user in the survey. Table 4.7 shows the ratings selected by the users in percentages.

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| OrchiD is much harder | 8% | 4% | 32% | **40%** | 16% | ORchiD is much easier |

Table 4.7: User percentages for general ORchiD difficulty

### 4.4.7 Overall comments

The participants were given spaces to share their thoughts regarding the desktop application, the repository and the metadata representation. These are categorized and summarized as follows.

**Strengths and positive points**:

- The actual uploading is very easy
- Satisfactory

- Much less sophisticated than other interfaces
- The name ORchiD
- Simple
- Compatability with a wide range of formats
- Very usable
- Realized potential
- Drag and Drop feature
- Good idea
- encouraging for creators to upload content

**Weaknesses and negative points**:

- Too many optional fields
- Fields difficult to understand and tooltip takes too long to appear
- The fields are overly complicated for general use
- Open licensing options were expected and not found
- Easy until the optional metadata layout is seen
- Unlikely that educators would take the time to populate the resource
- The description of the fields is unclear
- Error messages need help in resolution. Just stating the error is not enough
- The survey needed more detail
- The value of the client application is not obvious

**Opinions, suggestions and other comments**:

- Support for Creative Commons licensing would be great
- Some of the fields should be under the required fields, like “Keyword”
- A broad metadata standard would be useful for search and discovery
- A user friendly manual defining the metadata fields would be better than the tooltip feature
- Automatic metadata extraction can make this much easier

- Saving offline work can be useful when connectivity is not available
- More understandable headings and drop-down menus
- Enlarge upload button
- An upload progress bar could be useful

## 4.5 Discussion

It was shown that a diverse sample population with different educational roles, as shown in Figure 4.2, was reached by the use of the Survey Design Section 4.3.2 put into action. Even though the response rate seemed a bit slow at the time it was not a surprising outcome. There was 51 registered repository users by the end of the experiment. Only 50% of these users tested ORchiD and completed the survey. From those users that completed the survey, 2 individuals did not have a running desktop application and there was no reason to be found.

The repository integration proved to be technically successful. All of the issues raised were either resolved or a certain workaround was needed as shown in Table 4.1. The metadata representation at the repository side was also mostly correct except for compound fields with inner compound fields that had to be presented as XML. This was also evaluated by the users and most of them agreed to all the statements made regarding the repository's representation of the items and fields as shown in Table 4.6.

The user evaluations for the statements describing ORchiD suggest that most users thought that ORchiD was easy and simple as shown in Table 4.4. The Drag&Drop was liked by 16 users who agreed and strongly agree with the statement. Some users also expressed it as a strength in their comments. On the other hand, the tooltip feature did not have similar success as most of the users either disagreed or skipped this statement and also expressed their dislike in the comments. The feature was taking too long and was not always helpful as suggested by the evaluations. Users also did not have problems describing their resources using ORchiD except for a few of them.

Both ORchiD and the repository conformed to the IMS LOM specifications in representing all the fields but it was obvious that IMS LOM is a large metadata set and its comprehensiveness was not appreciated by most of the users. It was also highly scrutinized in the user comments even though most of the users were able to describe their resources. The user evaluations of IMS LOM shown in Table 4.5 showed that the majority of users expected other fields. Even though 17 of the users agreed that they were happy with the details they were presented with. The user comments cleared this confusion as 10 users found the metadata standard difficult, too large and overwhelming.

It could be argued that ORchiD has proven possible simplification to the repository deposit from the user views. The metadata set used limited most of the users from describing their resources and has been proven to be mostly overwhelming and time consuming from the User comments in Section 4.4.7. Nevertheless, ORchiD succeeded in areas where other deposit systems mentioned in the literature were lacking. Unlike the Authoring Add-in for Word, ORchiD is capable of handling various file types and more complex digital objects not only Word files. Moreover, it was not developed for a specific group like how OA-RJ was built for publishers. Debating whether a more generic solution is better can be worth discussing.

40% and 16% of the users gave ORchiD a score of 4 out 5 and 5 out 5 respectively in being much easier than their previous experiences with resource deposits.

## 4.6 Summary

This chapter discussed the evaluation of the repository integration, metadata representation and ORchiD. A survey and user evaluation was presented. Some issues were faced when integrating the repository with the desktop application but were mostly resolved as pointed out in this chapter. The metadata representation had some limitations in implementation and usage. ORchiD was proven successful in simplifying the deposit process but some problems were identified by the users.

Chapter 5

# Conclusion

There is a continuous need for simpler solutions for Open Educational Resources. One problem is populating repositories. There are many sides to this problem but one reason is that content creators need simpler applications to share OER.

ORchiD, Open Resources Depositor, has proven to be such an application. The presentation of the Learning Object Metadata was realized as a weakness because most users found the metadata standard overwhelming and expressed their concerns in their comments. Presenting all the metadata as options proved to be a bad solution and directly affected the perception of the users towards ORchiD. To some extent the deposit process was made simpler and many of the users liked the experience and expressed potential for ORchiD.

The adaptation of the EPrints repository to accomodate learning objects was proved to be successful. Several issues were faced but a few structural changes made it possible. The repository was also able to represent metadata specifications for the resources.

Evaluation of the tool showed that users generally had a good experience using the desktop application. Overall, the feedback from users suggested that the tool was simpler to use than other tools that they were familiar with and could contribute towards simplifying and encouraging the deposit and sharing of OERs.

The evaluation brought some future improvements to light. It was found that ORchiD is open for improvement with many features suggested by users.

## Chapter 6
# Future Work

1. *Exploring a better solution for metadata entry and description of resources:* by selecting a self-archiving model for content creators to create their OER on their own the metadata creation process needs to be simplified further. An investigation in this particular area needs to be done. Keeping metadata rich for search and retrieval but simple for deposit is still a problem. LOM was expressed as being too large. Some of the fields were also hard to understand.

2. *Integrating automatic metadata extraction methods and simplifying the representation of the fields:* extracting technical metadata from files is technically easy. On the other hand, the process of extracting metadata from many complex resources is rather challenging but would simplify the creation of OERs if it was achieved even if only to some extent.

3. *Adding further management services:* the release of SWORDv2 allow and specify more facilities for applications not just deposit. ORchiD, with the integration of SWORDv2, could allow users or content creators to view their resources, update and delet their content. It would decrease the direct interaction of the content creator with the repository.

4. *Enabling open licensing from the desktop application:* Open licensing is important for OER and enabling the user to add licenses to their resources from the application would simplify the deposit further.

5. *Integrate social networking with ORchiD to support collaborative authoring:* one of the strengths of OER is that it opens opportunities for collaborative work. Authors and content creators can work together to create better resources with actual contact. Versioning the same resources would also be useful. Sharing resources in different social networks from the same point of deposit can help in dissemination.

6. *Analyzing and comparing digital repository systems specifically for OER:* as it is difficult to select one repository system for a specific set of requirements, it may be useful to explore and analyze how other repository systems handle OER.

# Bibliography


[Abelson 2008] Hal Abelson. *The Creation of OpenCourseWare at MIT*. Journal of Science Education and Technology, vol. 17, no. 2, pages 164–174, 2008. (Cited on page 7.)

[AICC 2006] AICC. *Guidelines for Package Exchange Notification Services*. AICC, March 2006. (Cited on page 25.)

[Al-Khalifa & Davis 2006] Hend S. Al-Khalifa and Hugh C. Davis. *The evolution of metadata from standards to semantics in E-learning applications*. In Proceedings of the seventeenth conference on Hypertext and hypermedia, HYPERTEXT '06, pages 69–72, New York, NY, USA, 2006. ACM. (Cited on page 12.)

[Albright 2005] P. Albright. *Final Forum Report*. Rapport technique, UNESCO International Institute for Educational Planning, 2005. (Cited on page 8.)

[Allinson *et al.* 2008a] Julie Allinson, Les Carr, Jim Downing, David F. Flanders, Sebastien Francois, Richard Jones, Stuart Lewis, Martin Morrey, Glen Robson and Neil Taylor. *SWORD AtomPub Profile version 1.3*, 2008. (Cited on page 22.)

[Allinson *et al.* 2008b] Julie Allinson, Sebastien Francois and Stuart Lewis. *SWORD: Simple Web-service Offering Repository Deposit*. Ariadne, vol. 54, no. 54, pages 1–10, 2008. (Cited on pages 3 and 21.)

[Anido 2006] L. Anido. *An observatory for e-learning technology standards*. Adv. Technol. Learn., vol. 3, pages 99–108, April 2006. (Cited on page 12.)

[Atkins *et al.* 2007] Daniel E Atkins, John Seely Brown and Allen L Hammond. *A Review of the Open Educational Resources (OER) Movement: Achievements, Challenges, and New Opportunities*. Review Literature And Arts Of The Americas, vol. 2008, no. Book, Whole, page 84, 2007. (Cited on pages 1, 5 and 8.)

[Baker 2005] J. Baker. Introduction to open educational resources. Connexions, 2005. (Cited on page 6.)

[Barkman *et al.* 2002] Patricia Barkman, Debbie Brown, Peter Brusilovsky, J Richard Burke, Mike Fore, Jack Hyde, Ian Kegel, Bill Melton, Boyd Nielsen and Bruce Peoples. *Draft Standard for Learning Object Metadata*. Learning Technology Standards Committee of the IEEE, vol. 1484.12.1, no. July, pages 1–44, 2002. (Cited on pages 13, 16 and 17.)

[Baumgartner *et al.* 2007] Peter Baumgartner, Viola Naust, Agusti Canals, Nuria Ferrer, Julia Minguillon, Mireia Pascual, Mats Rajalakso, Timo Valiharju, Wernher Behrendt, Andreas Gruber, Veronika Prahauser and Sebastian Schaffert. *Open Educational Practices and Resources - OLCOS Roadmap 2012*. pages 87–109, 2007. (Cited on page 2.)

[Bekaert 2003] Jeroen Bekaert. *Using MPEG-21 DIDL to Represent Complex Digital Objects in the Los Alamos National Laboratory Digital Library*. D-Lib Magazine, vol. 9, no. 11, 2003. (Cited on page 20.)

[Berry 1999] R.S.Y. Berry. *Collecting data by in-depth interviewing*. 1999. (Cited on page 30.)

[Bradford *et al.* 2007] Peter Bradford, Margaret Porciello, Nancy Balkon and Debra Backus. *The Blackboard Learning System: The Be All and End All in Educational Instruction?* Journal of Educational Technology Systems, vol. 35, no. 3, pages 301–314, 2007. (Cited on page 28.)

[Buchanan *et al.* 2005] George Buchanan, David Bainbridge, Katherine J. Don and Ian H. Witten. *A new framework for building digital library collections*. In Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries, JCDL '05, pages 23–31, New York, NY, USA, 2005. ACM. (Cited on page 10.)

[Butcher 2011] Neil Butcher. A basic guide to open educational resources (oer). COL, UNESCO. Commonwealth of Learning, 2011. (Cited on page 1.)

[Caswell *et al.* 2008] Tom Caswell, Shelley Henson, Marion Jensen and David Wiley. *Open Content and Open Educational Resources: Enabling universal education*. The International Review of Research in Open and Distance Learning, vol. 9, no. 1, 2008. (Cited on page 8.)

[Cathro 2009] W. Cathro. *Metadata: An overview*. National Library of Australia Staff Papers, no. 0, 2009. (Cited on page 13.)

[DCMI 2005] DCMI. *Using Dublin Core - Dublin Core Qualifiers*. Website, November 2005. (Cited on page 14.)

[Dougiamas & Taylor 2003] Martin Dougiamas and Peter Taylor. *Moodle: Using Learning Communities to Create an Open Source Course Management System*. In David Lassner and Carmel McNaught, editeurs, Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2003, pages 171–178, Honolulu, Hawaii, USA, 2003. AACE. (Cited on page 23.)

[Duranceau & Rodgers 2010] E. F. Duranceau and R. Rodgers. *Automated IR deposit via the SWORD protocol: an MIT/BioMed Central experiment*. Serials, vol. 23, 2010. (Cited on page 27.)

[Duval *et al.* 2006] Erik Duval, N. Smith and M. Van Coillie. *Application profiles for learning.* In Proceedings of the Sixth International Conference on Advanced Learning Technologies, pages 242–246. IEEE, IEEE, 2006. (Cited on page 19.)

[EDiNA 2009] EDiNA. *EDiNA: Open Access Repository Junction.* Website, 2009. http://edina.ac.uk/projects/oa-rj/index.html. (Cited on pages 2 and 26.)

[EPrints 2011a] EPrints. *EPrints Manual.* Website, 2011. http://wiki.eprints.org/w/EPrints_Manual/. (Cited on page 10.)

[EPrints 2011b] EPrints. *EPrints software.* Website, 2011. http://www.eprints.org/software/. (Cited on page 10.)

[Fielding *et al.* 1997] R. Fielding, J. Gettys, J. Mogul, H. Frystyk and T. Berners-Lee. *Hypertext Transfer Protocol – HTTP/1.1.* RFC 2068 (Proposed Standard), jan 1997. Obsoleted by RFC 2616. (Cited on page 21.)

[Fletcher 2012] Kathi Fletcher. *OER Publishing API: SWORD for OER.* Website, January 2012. (Cited on page 27.)

[Friesen *et al.* 2002] Norm Friesen, Anthony Roberts and Sue Fisher. *CanCore: Metadata for Learning Objects.* Canadian Journal of Learning and Technology, vol. 28, no. 3, pages 43–53, 2002. (Cited on page 19.)

[Greenberg 2005] Jane Greenberg. *Understanding Metadata and Metadata Schemes.* Cataloging Classification Quarterly, vol. 40, no. 3, pages 17–36, 2005. (Cited on page 15.)

[Gregorio & De hOra 2007] J. Gregorio and B. De hOra. *The Atom Publishing Protocol.* RFC 5023 (Proposed Standard), 2007. (Cited on pages 21 and 22.)

[Guha 2008] Tamal Kumar Guha. *Dublin Core: the standard for networking libraries.* E-prints in Library & Information Science (e-Lis), 2008. (Cited on page 14.)

[Haase 2004] Kenneth Haase. *Context for semantic metadata.* In Proceedings of the 12th annual ACM international conference on Multimedia, MULTIMEDIA '04, pages 204–211, New York, NY, USA, 2004. ACM. (Cited on page 12.)

[Hatala *et al.* 2004a] Marek Hatala, Griff Richards, Timmy Eap and Jordan Willms. *The EduSource Communication Language: Implementing Open Network for Learning Repositories and Services.* In in Special Track on Engineering e-Learning Systems held at ACM Symposium on Applied Computing (SAC) 2004, pages 957–962. ACM Press, 2004. (Cited on page 25.)

[Hatala *et al.* 2004b] Marek Hatala, Griff Richards, Timmy Eap and Jordan Willms. *The interoperability of learning object repositories and services: standards,* *implementations and lessons learned*. In Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, WWW Alt. '04, pages 19–27, New York, NY, USA, 2004. ACM. (Cited on page 12.)

[Hatala *et al.* 2004c] Marek Hatala, Griff Richards, Scott Thorne and Jeff Merriman. *Closing the Interoperability Gap: Connecting Open Service Interfaces with Digital Repository Interoperability*. In Lorenzo Cantoni and Catherine McLoughlin, editeurs, Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2004, pages 78–83, Lugano, Switzerland, 2004. AACE. (Cited on page 25.)

[Hayes 2005] Helen Hayes. *Digital Repositories - Helping universities and colleges - Briefing Paper*. JISC, 2005. (Cited on page 8.)

[Heery & Patel 2000] Rachel Heery and Manjula Patel. *Application profiles: mixing and matching metadata schemas*. Ariadne, no. 25, SEP 2000. (Cited on pages 13 and 15.)

[Hewlett 2005] Hewlett. *Open Educational Resources Initiative Overview*, 2005. (Cited on page 7.)

[Howell & Stuart 2009] Fred Howell and Ian Stuart. *EM-Loader Project Documentation*. Website, 2009. http://publicationslist.org/em-loader/emloader-report-intro.html. (Cited on page 26.)

[IMS 2003a] IMS. *IMS Content Packaging Best Practices and Implementstion Guide*. IMS Global Learning Consortium, version 1.1.3 édition, jun 2003. (Cited on page 19.)

[IMS 2003b] IMS. *IMS Content Packaging Information Model*. IMS Global Learning Consortium, version 1.1.3 édition, jun 2003. (Cited on page 19.)

[IMS 2003c] IMS. *IMS Content Packaging XML Binding*. IMS Global Learning Consortium, version 1.1.3 édition, jun 2003. (Cited on page 19.)

[IMS 2003d] IMS. *IMS Digital Repositories Interoperability - Core Functions Information Model Version 1.0 Final Specification*, 2003. (Cited on pages 19 and 25.)

[IMS 2006] IMS. *IMS Metadata Best Practice Guide for IEEE 1484.12.1-2002 Standard for Learning Object Metadata*. IMS Global Learning Consortium, version 1.3 édition, aug 2006. (Cited on pages 3, 12 and 19.)

[Jones 2011] Richard Jones. *SWORD 2.0 Profile*, 2011. (Cited on page 23.)

[Kavcic 2011] A. Kavcic. *Implementing Content Packaging Standards*. In EUROCON - International Conference on Computer as a Tool (EUROCON), 2011 IEEE, april 2011. (Cited on page 21.)

[Kunze & Baker 2007] J. Kunze and T. Baker. *The Dublin Core Metadata Element Set*. RFC 5013 (Informational), August 2007. (Cited on page 13.)

[Lagoze *et al.* 2005] Carl Lagoze, Sandy Payette, Edwin Shin and Chris Wilper. *Fedora: An Architecture for Complex Objects and their Relationships*. CoRR, vol. abs/cs/0501012, 2005. (Cited on pages 8 and 9.)

[Learning 2011] Advanced Distributed Learning. *Sharable Content Object Reference Model*. Website, 2011. (Cited on page 20.)

[Lewis *et al.* 2009] Stuart Lewis, Leonie Hayes, Vanessa Newton-Wade, Antony Corfield, Richard Davis, Tim Donohue and Scott Wilson. *If SWORD is the answer, what is the question?: Use of the Simple Web-service Offering Repository Deposit protocol*. Program Electronic Library And Information Systems, vol. 43, no. 4, pages 407–418, 2009. (Cited on page 22.)

[Lewis *et al.* 2012] Stuart Lewis, Pablo de Castro and Richard Jones. *SWORD: Facilitating Deposit Scenarios*. The Magazine of Digital Library Research, vol. 18, no. 1/2, 2012. (Cited on page 21.)

[Lewis 2011] Stuart Lewis. *EasyDeposit - The SWORD client creation tool*. Website, 2011. http://easydeposit.swordapp.org/instructions/. (Cited on page 27.)

[Lihitkar & Lihitkar 2012] S.R. Lihitkar and R.S. Lihitkar. *Open Source Software for Developing Digital Library: Comparative Study*. DESIDOC Journal of Library & Information Technology, vol. 32, no. 5, 2012. (Cited on pages 11 and 12.)

[Likert 1932] R. Likert. *A technique for the measurement of attitudes*. Archives of psychology, 1932. (Cited on page 53.)

[Lukasiak *et al.* 2004] Jason Lukasiak, Shirley Agostinho, Ian Burnett, Gerrard Drury, Jason Goodes, Sue Bennett, Lori Lockyer and Barry Harper. *A Framework for the Flexible Content Packaging of Learning Objects and Learning Designs*. Journal of Educational Multimedia and Hypermedia, vol. 13, no. 4, pages 465–481, October 2004. Special Issue: Special Issue. (Cited on pages 19 and 20.)

[Maly *et al.* 2004a] K. Maly, M. Nelson, M. Zubair, A. Amrou, S. Kothamasa, L. Wang and R. Luce. *Kepler - A Communal Digital Library*, 2004. (Cited on page 27.)

[Maly *et al.* 2004b] Kurt J. Maly, Michael L. Nelson, Mohammad Zubair, Ashraf Amrou, S. Kothasama, Lan Wang and Richard Luce. *Light-weight communal digital libraries*. In Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries, JCDL, pages 237–238, New York, NY, USA, 2004. ACM. (Cited on page 27.)

[Mason & Ip 1998] Jon Mason and Albert Ip. *EdNA, HIGHER EDUCATION AND METADATA IMPLEMENTATION*, 1998. (Cited on page 15.)

[Massart 2007] David Massart. *The EUN Learning Resource Exchange (LRE)*. In The 15th International Conference on Computers in Education 2007, pages 170–174, 2007. (Cited on page 24.)

[McClelland 2003] M. McClelland. *Metadata standards for educational resources*. Computer, vol. 36, no. 11, pages 107 – 109, nov 2003. (Cited on pages 3 and 13.)

[McGill *et al.* 2008] L. McGill, S. Currier, C. Duncan and P. Douglas. *Good intentions: improving the evidence base in support of sharing learning materials*. Rapport technique, JISC, 2008. (Cited on page 2.)

[Morgan 2004] Eric L. Morgan. *An Introduction to the Search/Retrieve URL Service (SRU)*. Ariadne, no. 40, July 2004. (Cited on page 25.)

[Murray 1999] P. Murray. *Fundamental issues in questionnaire design*. Accident and emergency nursing, vol. 7, no. 3, pages 148–153, 1999. (Cited on page 52.)

[Nielsen 1997] J. Nielsen. *The use and misuse of focus groups*. Software, IEEE, vol. 14, no. 1, pages 94–95, 1997. (Cited on page 30.)

[Nottingham & Sayre 2005] Mark Nottingham and Robert Sayre. *RFC 4287 The Atom Syndication Format*. Rapport technique, Internet Engineering Task Force (IETF), dec 2005. (Cited on page 21.)

[Pinfield *et al.* 2002] Stephen Pinfield, Mike Gardner and John MacColl. *Setting up an institutional e-print archive*. Ariadne, 2002. (Cited on page 8.)

[Prause *et al.* 2007] Christian R. Prause, Stefaan Ternier, Tim de Jong, Stefan Apelt, Marius Scholten, Martin Wolpers, Markus Eisenhauer, Bram Vandeputte, Marcus Specht and Erik Duval. *Unifying Learning Object Repositories in MACE*. In Frans van Assche, editeur, Proceedings of the First International Workshop on Learning Object Discovery & Exchange (LODE'07), 2007. (Cited on page 25.)

[Pyrounakis & Nikolaidou 2009] G. Pyrounakis and M. Nikolaidou. *Comparing Open Source Digital Library Software*. Handbook of Research on Digital Libraries: Design, Development, and Impact, pages 51–60, 2009. (Cited on page 11.)

[Research 2011] Microsoft Research. *Microsoft Article Authoring Add-in User Guide*. Website, January 2011. (Cited on pages 2 and 26.)

[Roy *et al.* 2010] Devshri Roy, Sudeshna Sarkar and Sujoy Ghose. *A Comparative Study of Learning Object Metadata, Learning Material Repositories, Metadata Annotation & an Automatic Metadata Annotation Tool*. tmrfindiaorg, vol. 2, pages 103 – 126, 2010. (Cited on pages 15 and 19.)

[Sierra *et al.* 2005] Jose Luis Sierra, Pablo Morenoger and Ivan Martinezortiz. Building learning management systems using ims standards: Architecture of a manifest driven approach 1 the ims content packaging and the concept of manifest, pages 1–12. Springer, 2005. (Cited on page 19.)

[Smith *et al.* 2003] MacKenzie Smith, Mary Barton, Mick Bass, Margret Branchofsky, Greg McClellan, Dave Stuve, Robert Tansley and Julie Harford Walker. *DSpace An Open source Dynamic Digital Repository*. D-Lib Magazine, vol. 9, no. 1, 2003. (Cited on page 9.)

[SPI 2010] SPI. *The Simple Publishing Interface (SPI) Specification*. Rapport technique, European Committee for Standardization, Feb 2010. (Cited on page 24.)

[Staples *et al.* 2003] T. Staples, R. Wayland and S. Payette. *The Fedora Project*. The Magazine of Digital Library Research, vol. 9, no. 4, 2003. (Cited on page 8.)

[Suleman 2006] Hussein Suleman. *Reflections on Three Years of Archiving Research Output*. Rapport technique, UCT CS Research Document Archive, 2006. (Cited on page 31.)

[Tansley *et al.* 2003] Robert Tansley, Mick Bass, David Stuve, Margret Branschofsky, Daniel Chudnov, Greg McClellan and MacKenzie Smith. *The DSpace institutional digital repository system: current functionality*. In Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries, JCDL '03, pages 87–97, Washington, DC, USA, 2003. IEEE Computer Society. (Cited on pages 8, 9 and 21.)

[Tarrant *et al.* 2012] David Tarrant, Steve Hitchcock, Richard Jones, Tim Brody, David Flanders, Mark MacGillvray and Ian Stuart. *DeposiMO Profile*, 2012. (Cited on page 26.)

[Ternier & Duval 2003] Stefaan Ternier and Erik Duval. *Web services for the ARIADNE Knowledge Pool System*. In 3rd Annual Ariadne Conference, pages 1–9, 2003. (Cited on page 25.)

[Ternier *et al.* 2008] Stefaan Ternier, David Massart, Frans Van Assche, Neil Smith, Bernd Simon and Erik Duval. *A Simple Publishing Interface For Learning Object Repositories*. In Joseph Luca and Edgar R. Weippl, editeurs, Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2008, pages 1840–1845, Vienna, Austria, June 2008. AACE. (Cited on page 23.)

[Ternier *et al.* 2010] Stefaan Ternier, David Massart, Michael Totschnig, Joris Klerkx and Erik Duval. *The Simple Publishing Interface*. The Magazine of Digital Library Research, vol. 16, no. 9/10, Sep 2010. (Cited on page 24.)

[Tuomi 2006] I. Tuomi. *Open Education Resources: What they are and why do they matter*, 2006. (Cited on page 6.)

[UNESCO 2002] UNESCO. *Forum on the Impact of Open Courseware for Higher Education in Developing Countries*. Rapport technique, William and Flora Hewlett Foundation, 2002. (Cited on page 1.)

[Verbert & Duval 2007] Katrien Verbert and Erik Duval. *Evaluating the ALOCOM Approach for Scalable Content Repurposing*. In Erik Duval, Ralf Klamma and Martin Wolpers, editeurs, EC-TEL, volume 4753 of *Lecture Notes in Computer Science*, pages 364–377. Springer, 2007. (Cited on page 24.)

[Warr & Hangsing 2009] H. Warr and D.P. Hangsing. *Open Source Digital Library Software: A Literature Review*. In Proceedings of the National Seminar on'Preservation and Conservation of Information Resources in Knowledge Society: Issues, Challenges and Trends, pages 238–258, 2009. (Cited on page 11.)

[Weibel 2010] S.L. Weibel. *Dublin Core Metadata Initiative (DCMI): A Personal History*. Encyclopdia of Library and Information Sciences, pages 1655–1663, 2010. (Cited on page 13.)

[Wells 2010] Stephanie Wells. *DURA - Direct User Repository Access*. JISC, 2010. (Cited on page 27.)

[Wiley 2006] David Wiley. *On the Susainability of Open Educational Resources in Higher Education*. Higher Education, vol. 26, no. 7, page 21, 2006. (Cited on page 8.)

[Witten *et al.* 2000] Ian H. Witten, Stefan J. Boddie, David Bainbridge and Rodger J. McNab. *Greenstone: a comprehensive open-source digital library software system*. In Proceedings of the fifth ACM conference on Digital libraries, DL '00, pages 113–121, New York, NY, USA, 2000. ACM. (Cited on pages 8 and 10.)

[Yuan *et al.* 2008] Li Yuan, Sheila MacNeill and Wilbert Kraan. *Open Educational Resources - Opportunities and Challenges for Higher Education*. JISC CETIS OER Briefing Paper, pages 1–34, 2008. (Cited on pages 1 and 6.)

## Appendix A

# Appendix 1

## A.1 Survey Post

MSc CS project @:

Open Educational Resources Depositor

ORchiD Evaluation Survey

Dear Content Creator

You are invited to participate in the research project ***An End-to-end Solution for Complex Open Educational Resources*** by trying out the desktop tool (ORchiD) and completing a questionnaire. The importance and potential impact of Open Educational Resources is being realized by many organizations and individuals around the world. There is a continuous need for simpler and more effective approaches to assist in this movement. In turn, many projects focusing on different aspects of OERs arose throughout the past decade or so...

ORchiD focuses on simplifying the deposit process by minimizing the interaction between the “Content Creator” and the repository using the desktop as the main workspace.

All of the information you need is presented to you in the survey including the download links for the application. You will also be presented with an **informed consent** at the first stage of the survey. I appreciate your attention and participation.

Please note that you may *save* your survey and *resume* at any point. The survey in itself is short but the tasks may take around 40 minutes depending on your resource description details. You may also contact me if you have any questions or comments on this email. Click on the link below to go to the survey page.

Survey link: http://banzai.cs.uct.ac.za/survey/index.php?sid=91398

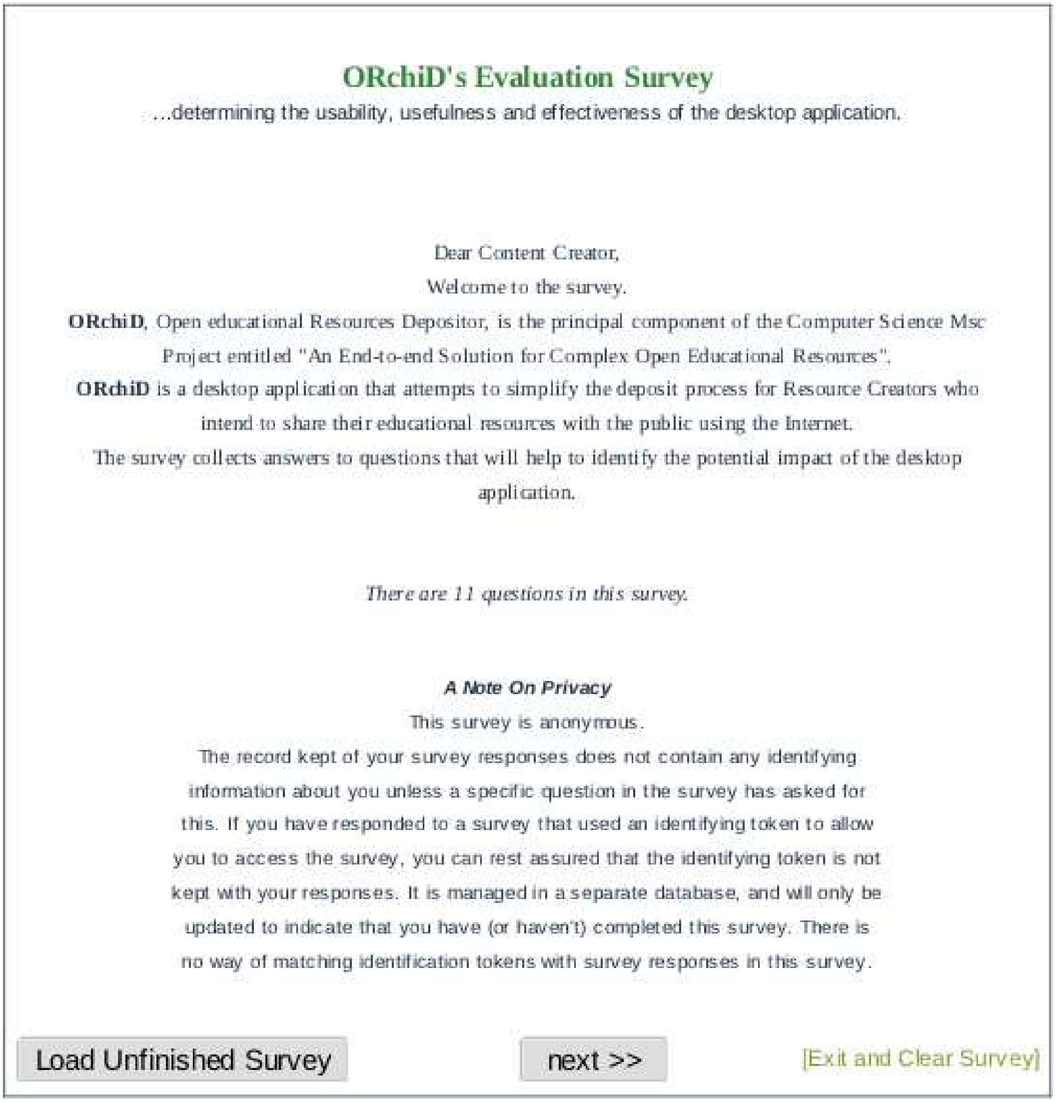
Best regards,

*Morwan Mohamed Nour*

================================
Digital Libraries Laboratory
Postgraduate Lab
Room 300 Computer Science Building
18 University Avenue
University of Cape Town
================================

## A.2 Survey screen

### Survey Welcome



ORchiD's Evaluation Survey

...determining the usability, usefulness and effectiveness of the desktop application.

Dear Content Creator,

Welcome to the survey.

**ORchiD**, Open educational Resources Depositor, is the principal component of the Computer Science Msc Project entitled "An End-to-end Solution for Complex Open Educational Resources".

**ORchiD** is a desktop application that attempts to simplify the deposit process for Resource Creators who intend to share their educational resources with the public using the Internet.

The survey collects answers to questions that will help to identify the potential impact of the desktop application.

*There are 11 questions in this survey.*

***A Note On Privacy***

This survey is anonymous.

The record kept of your survey responses does not contain any identifying information about you unless a specific question in the survey has asked for this. If you have responded to a survey that used an identifying token to allow you to access the survey, you can rest assured that the identifying token is not kept with your responses. It is managed in a separate database, and will only be updated to indicate that you have (or haven't) completed this survey. There is no way of matching identification tokens with survey responses in this survey.

Load Unfinished Survey | next >> | [Exit and Clear Survey]

## Stage 1: Introduction and Consent

**ORchiD's Evaluation Survey**

...determining the usability, usefulness and effectiveness of the desktop application.

0% [ ] 100%

**Informed Consent**

**Researcher:**

NAME: **Morwan** Mohamed Nour

STUDENT NUMBER: MHMMOR001

CONTACT: mnour@cs.uct.ac.za OR morwan.nour@gmail.com

**Supervisor:**

NAME: Hussein Suleman

CONTACT: hussein@cs.uct.ac.za

**MSc Computer Science Project:**

TITLE: An End-to-end Solution for Complex Open Educational Resources

AIM: Attempt to simplify depositing into repositories using a desktop application

**1 -- INTRODUCTORY STATEMENT**

Open Educational Resources can be defined as digital objects of any type that have any learning impact on anyone shared "freely" using the Internet.

We developed an application (ORchiD) that creates a bundle of user-chosen resources with user-entered annotations and descriptions that is then deposited to the repository. During this survey you (the user) will attempt to perform tasks which will be explained to you as you go on.

You are invited to participate in this survey because of your expertise in content creation or dissemination and your possible openness to educational content sharing. All the necessary information is explained to you. If at any point you have any queries or comments please contact me (**Morwan**) via email.

**2 -- PURPOSE**

This survey should help analyze and evaluate ORchiD.

**3 -- TASKS**

If you decide to consent and participate you will be asked to:

- Create a user by registering at ORchiD's Repository
- Download and Run ORchiD
- Deposit a single file Educational Resource
- Deposit a multi-file Educational Resource
- Answer a few questions during and after performing the tasks

*****These tasks will be detailed on the next stages of the survey***

**4 -- ENVIRONMENT & DURATION**

You will perform the tasks wherever you please. The point is to protray real-life scenarios like "a lecturer willing to share his/her lecture slides at the end of a lecture" or "a course convenor trying to share his/her course resources at the end of a semester". Completion of the tasks and the survey will be user-dependent and may take around 40 minutes.

**5 -- CONFIDENTIALITY**

If you consent to participate in this evaluation, your personal information will be kept confidential. Any information you provide in the survey or the test repository is kept confidential between you and the researcher. You may also request the removal of any items you deposit.

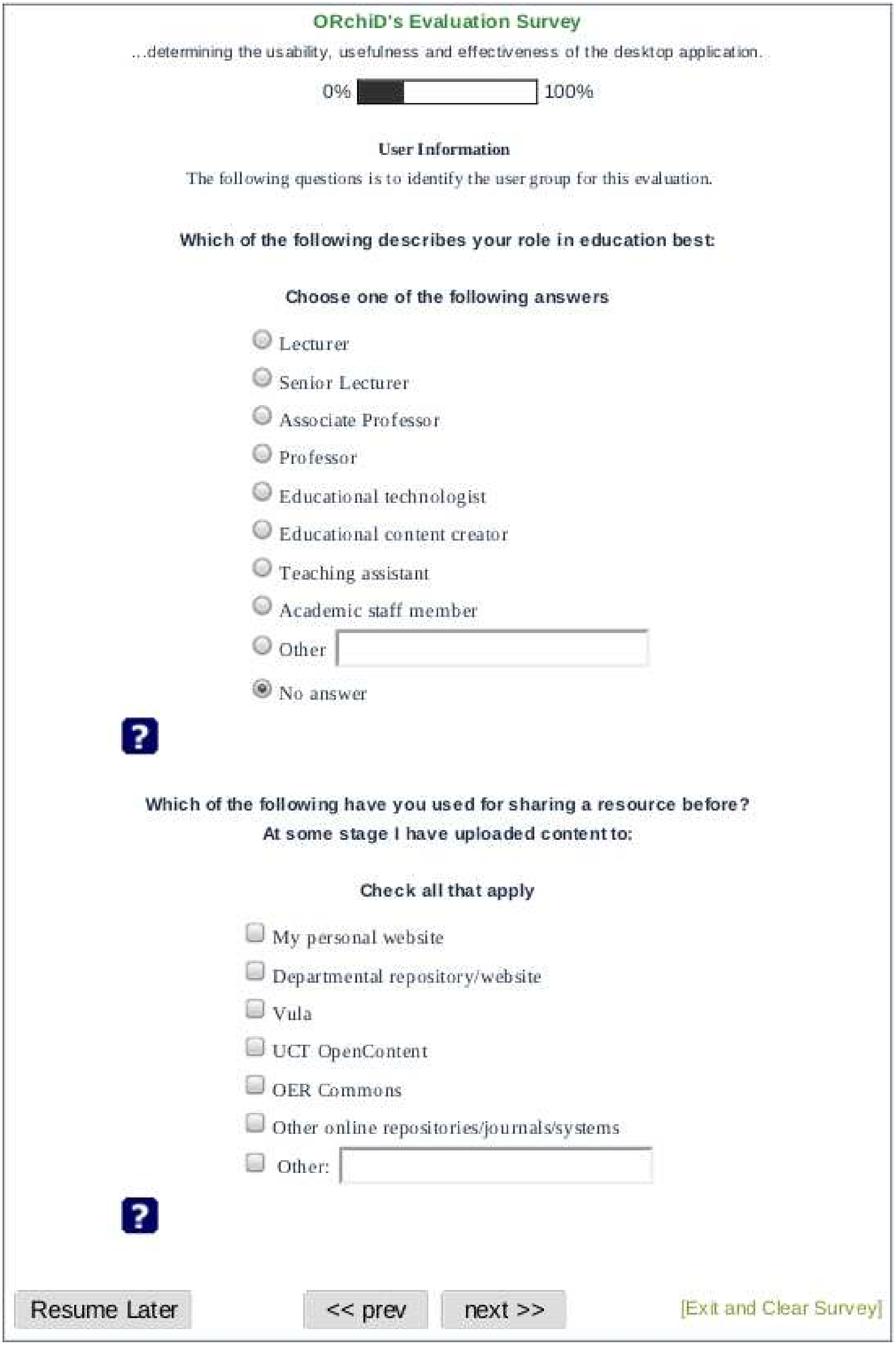
**6 -- STATEMENT OF CONSENT**

"I acknowledge that I have read the above explanation of this evaluation. I understand that the collected data from this survey will be analyzed and used to evaluate the mentioned desktop application. I also understand that the researcher will not disclose my personal information. By selecting the option 'yes' below I agree to participate in this evaluation."

*"In advance, thank you for your precious time and effort." --Morwan*


Yes
No
Select yes to agree and continue.
***Note: if you have any further questions or concerns please contact Morwan at
(morwan.nour@gmail.com)
Resume Later
<< prev
next >>
[Exit and Clear Survey]

### Stage 2: User Information



ORchiD's Evaluation Survey

...determining the usability, usefulness and effectiveness of the desktop application.

0% 100%

**User Information**

The following questions is to identify the user group for this evaluation.

**Which of the following describes your role in education best:**

**Choose one of the following answers**

- Lecturer
- Senior Lecturer
- Associate Professor
- Professor
- Educational technologist
- Educational content creator
- Teaching assistant
- Academic staff member
- Other
- No answer

?

**Which of the following have you used for sharing a resource before?**
**At some stage I have uploaded content to:**

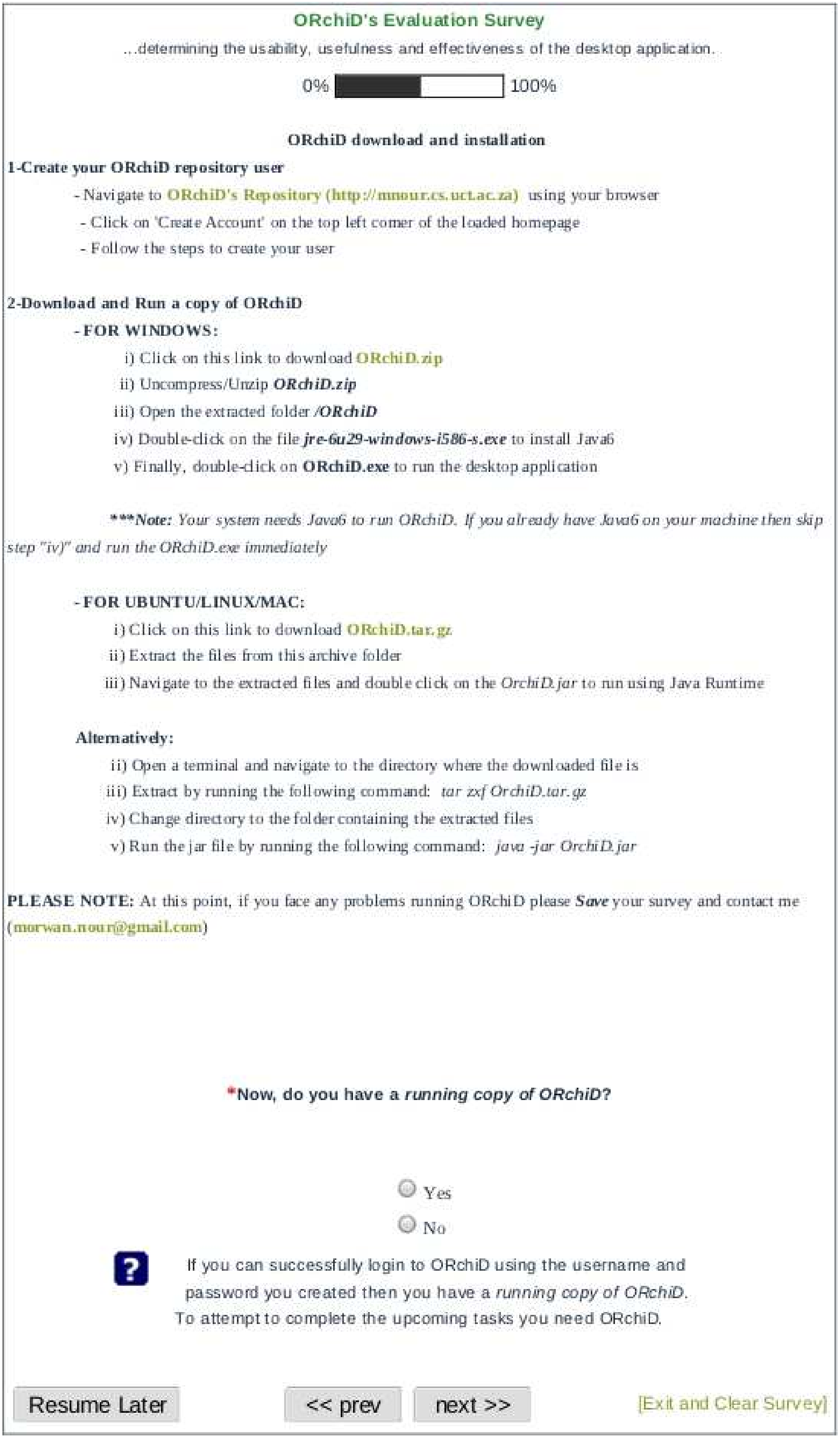
**Check all that apply**

- My personal website
- Departmental repository/website
- Vula
- UCT OpenContent
- OER Commons
- Other online repositories/journals/systems
- Other:

?

Resume Later | << prev | next >> | [Exit and Clear Survey]

### Stage 3: ORchiD download and installation



ORchiD's Evaluation Survey

...determining the usability, usefulness and effectiveness of the desktop application.

0% 100%

**ORchiD download and installation**

**1-Create your ORchiD repository user**

- Navigate to ORchiD's Repository (http://mnour.cs.uct.ac.za) using your browser
- Click on 'Create Account' on the top left corner of the loaded homepage
- Follow the steps to create your user

**2-Download and Run a copy of ORchiD**

**- FOR WINDOWS:**

i) Click on this link to download ORchiD.zip

ii) Uncompress/Unzip ***ORchiD.zip***

iii) Open the extracted folder ***/ORchiD***

iv) Double-click on the file ***jre-6u29-windows-i586-s.exe*** to install Java6

v) Finally, double-click on **ORchiD.exe** to run the desktop application

******Note:*** *Your system needs Java6 to run ORchiD. If you already have Java6 on your machine then skip step "iv)" and run the ORchiD.exe immediately*

**- FOR UBUNTU/LINUX/MAC:**

i) Click on this link to download ORchiD.tar.gz

ii) Extract the files from this archive folder

iii) Navigate to the extracted files and double click on the *OrchiD.jar* to run using Java Runtime

**Alternatively:**

ii) Open a terminal and navigate to the directory where the downloaded file is

iii) Extract by running the following command: *tar zxf OrchiD.tar.gz*

iv) Change directory to the folder containing the extracted files

v) Run the jar file by running the following command: *java -jar OrchiD.jar*

**PLEASE NOTE:** At this point, if you face any problems running ORchiD please ***Save*** your survey and contact me (morwan.nour@gmail.com)

*Now, do you have a *running copy of ORchiD*?

Yes

No

If you can successfully login to ORchiD using the username and password you created then you have a *running copy of ORchiD*. To attempt to complete the upcoming tasks you need ORchiD.

Resume Later | << prev | next >> | [Exit and Clear Survey]

## Stage 4: Tasks and questions

**ORchiD's Evaluation Survey**

...determining the usability, usefulness and effectiveness of the desktop application.

0% 100%

**Depositing educational resources using ORchiD**

**SOME TIPS BEFORE STARTING**

- DRAG & DROP Feature: you can drag files or folder into the white box when adding your resources
- TOOLTIPS: you can roll your mouse over a field name to get a better description about the field.
- OPTIONAL fields: these extra fields and categories (top-right dropdown box in the "Optional Details" tab) are there to help you describe your resources more. You may enter as little or as much data as you like.
- KNOWN issue: If your upload fails try logging out and logging back in to the application. The log out button is in the "File" menu. Your anti-virus or OS security module may have prevented the application from storing your details.

*****First Task: Depositing a single file resource**

Examples: Lecture slides, Assignment, image, Conference paper, Instructional video, etc.

Follow the steps below to complete the **First Task**:

1- Run ORchiD
2- Log in to the desktop application using the username and password you created in the repository website
3- Browse and select the file you want to share by clicking on the "Add file" button
4- Type in the "Title" and "Description" fields
5- Click on the "Optional Details" tab to add more "metadata" to your resource
6- Click on the "Upload" button on the "Main" tab window

*****Second Task: Depositing a multi-file resource**

Examples: Course101 term resources, MyBook chapters, Event proceedings with photos, etc.

Follow the same steps as the **First Task** but this time use a multiple file resource (collection).
You will try to describe your collection as a single entity and then click the upload button.

*****Third Task: Viewing your resources in the repository**

1- Go back to ORchiD's repository website (mnour.cs.uct.ac.za) where you created your user at the previous stage.
2- View your OERs in the Live Repository or Login and view your resources.

* **Have you completed the tasks successfully?**

**Choose one of the following answers**

○ Yes
○ Only the First task was a success.
○ Only the Second task was a success.
○ No

? At a successful upload a "**Success: item(s) deposited to the repository!**" will appear on the status bar found in the bottom of the "Main" window

**1: Strongly Disagree**
**2: Disagree**
**3: Neutral**
**4: Agree**
**5: Strongly Agree**

| | 1 | 2 | 3 | 4 | 5 | No answer |
|---|---|---|---|---|---|---|
| I immediately **understood** what I needed to do. | ○ | ○ | ○ | ○ | ○ | ◉ |
| The application was **easy** to work with. | ○ | ○ | ○ | ○ | ○ | ◉ |
| I **like** the drag&drop feature. | ○ | ○ | ○ | ○ | ○ | ◉ |
| I was able to **describe** my resource in the First Task. | ○ | ○ | ○ | ○ | ○ | ◉ |
| I was able to **describe** my resources in the Second Task. | ○ | ○ | ○ | ○ | ○ | ◉ |
| The optional fields were easy to **comprehend**. | ○ | ○ | ○ | ○ | ○ | ◉ |
| The tooltip feature was **useful**. | ○ | ○ | ○ | ○ | ○ | ◉ |
| In general, the application is **simple**. | ○ | ○ | ○ | ○ | ○ | ◉ |

**How would you rate the following statements about the category representations and the optional metadata fields**

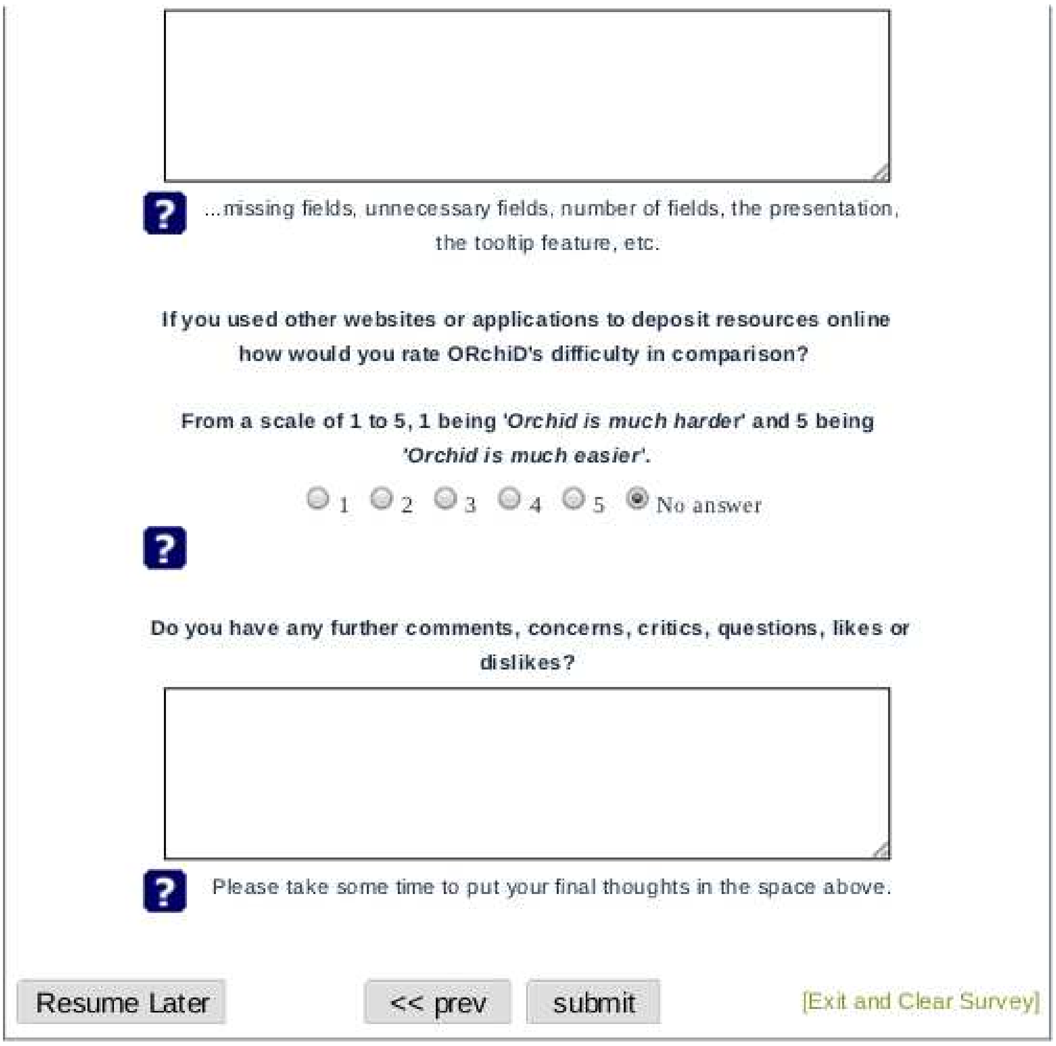
**1:Strongly Disagree**
**2:Disagree**

| | 1 | 2 | 3 | 4 | 5 | No answer |
|---|---|---|---|---|---|---|
| The fields are **suitable** to represent my resources. | ○ | ○ | ○ | ○ | ○ | ◉ |
| I expect other fields. Something is **missing**. | ○ | ○ | ○ | ○ | ○ | ◉ |
| More fields should be **required** not just the *title* and *description*. | ○ | ○ | ○ | ○ | ○ | ◉ |
| In general, I am **happy** with the details presented for resource description. | ○ | ○ | ○ | ○ | ○ | ◉ |

**How would you rate the following statements about the repository website regarding your current experience when viewing your items?**

**1: Strongly Disagree**
**2: Disagree**
**3: Neutral**
**4: Agree**
**5: Strongly Agree**

| | 1 | 2 | 3 | 4 | 5 | No answer |
|---|---|---|---|---|---|---|
| My items are **represented** correctly. | ○ | ○ | ○ | ○ | ○ | ◉ |
| The **presentation** of my items was as I expected. | ○ | ○ | ○ | ○ | ○ | ◉ |
| The information presented is easy to **understand**. | ○ | ○ | ○ | ○ | ○ | ◉ |


...missing fields, unnecessary fields, number of fields, the presentation, the tooltip feature, etc.
If you used other websites or applications to deposit resources online how would you rate ORchiD's difficulty in comparison?
From a scale of 1 to 5, 1 being 'Orchid is much harder' and 5 being 'Orchid is much easier'.
1 2 3 4 5 No answer
Do you have any further comments, concerns, critics, questions, likes or dislikes?
Please take some time to put your final thoughts in the space above.
Resume Later
<< prev
submit
[Exit and Clear Survey]