IR to RIMS: Transforming an Institutional Repository Into a Research Information Management System

CRAIG FELDMAN, University of Cape Town
DARRYL MEYER, University of Cape Town

1. PROJECT DESCRIPTION

The NRF has two databases running on legacy systems: the 'Current and Completed Research Projects' and the 'NRF Funded Projects'. These databases need to be migrated to the modern DSpace[1] digital repository system. As part of our honours project, we will migrate the two databases to DSpace and customise the DSpace user interface (UI) to fit the requirements of the NRF. We have identified and will be developing a set of unique tools and plugins that will be useful to the NRF and hopefully the broader DSpace community. By integrating these tools into DSpace, we hope to transform DSpace into a Research Information Management System (RIMS). An RIMS is used to store and manage the intellectual data created by an institution. The additions will include a tool to provide automatic and manual mapping of legacy data to the DSpace metadata fields, an ingestion manager that can ingest the mapped data into a DSpace repository and a report writing tool to generate managerial reports from the repository.

1.1 Project Significance

DSpace is a system that allows institutions to preserve and disseminate their intellectual works. According to the Registry of Open Access Repositories [ROAR 2015], DSpace is the most widely used digital repository system, therefore tools that are developed to be useful for DSpace are likely to be well received. The current solutions for migrating legacy data involves users having to understand how the legacy system stored the data, then creating custom scripts capable of formatting the data into a DSpace accepted format. The data would then be imported using the command line tool that DSpace provides. We want to provide automated tools to the users of DSpace to perform data migrations faster and easier, with reduced required user interaction. This will decrease the amount of time it takes to migrate legacy data to a DSpace repository and it will allow the migration to be performed by users who may be unfamiliar with the DSpace system. We also want to provide the ability to create formatted and customised reports based on the information available in the repository. These reports could be used for making decisions, developing summaries or other purposes. These features will help transform DSpace into an RIMS by including feature that are already available in RIMS software packages.

1.2 Project Issues and Difficulties

During the course of this project we will address the issue of ensuring that the software we develop is as generic as possible so that it can be useful to the widest possible group of users. To achieve this, we will have to test the software using data from a variety of different sources. Challenges specific to the automatic mapper may be that the field names of the legacy data provided cannot be determined. We will need to ensure that the methods used are capable of determining field names and mappings from a wide variety of inputs. For the report writer, it is likely we will modify and extend an existing open source report writing tool to integrate it with the DSpace repository. One of the main challenges to using an existing piece of software is the initial learning and understanding of the code that may have little to no documentation. There may also be incompatibility between what the existing open source tool requires and what the DSpace system provides.

We will be taking a user-centered design (UCD) approach when designing and developing these software tools. The issue that this raises is that the NRF is based in Pretoria and we are based in Cape Town. We can accommodate this by communicating via email but this limits the level of interaction between us and the NRF and increase the amount of time spent communicating.

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

2. PROBLEM STATEMENT

2.1 Aims and Research Question

This project has been requested by the NRF. Their initial requirement was to migrate two legacy database systems into a new DSpace repository. We aim to meet this requirement, as well as to provide additional features to the NRF that will prove useful and help to streamline the process of adding items into DSpace.

One of the main aims of our work, is to develop a tool that will facilitate the migration of data from a legacy database system into DSpace. We hope to develop a generic tool that will work for a wide range of inputs. The tool should simplify the process of setting up a DSpace repository from a previous system, by incorporating an automatic mapper to map the legacy fields to the DSpace Dublin Core metadata fields.

The second aspect of this project is to develop a DSpace plugin that would allow for the automatic creation of customised and formatted reports. These reports will likely prove useful to the NRF and other organisations.

2.2 Requirements

This project is mainly a Software Engineering project, however, we aim to add a significant and unique contribution to the community. We have not found a generic tool that can work for the migration from many different systems and previous projects of this nature have only focused on transferring from a specific system, as such the tools developed in these prior projects are not generic and serve only a one time purpose. Furthermore, we have not found a tool to facilitate the automatic mapping of fields. A DSpace report writing tool developed by @mire [2] exists; however the development is closed source and the software is not free.

The users of this new system will be the NRF, as well as universities throughout South Africa. We will use our created tool to migrate the NRF's current system into a new DSpace repository. Universities will then be able to add entries directly into the DSpace repository, pending the additions' approval by an NRF staff member. This will greatly streamline the process of adding items, as currently universities send information about masters and PhD projects to the NRF, who then have to manually add this into their database. We hope that our system will be adopted by other organisations in the future.

All features must be thoroughly tested and should work given a variety of use cases. The project requirements are discussed in more detail throughout this proposal.

3. PROCEDURES AND METHODS

3.1 Development Procedures

The procedure for our project will be to migrate the legacy data, provided by the NRF, to a new and customized DSpace instance running on the servers located at the NRF. We will use the tools we create to ingest the legacy data. The development of the tools will start with the automatic and manual field mapper, then, once we have a standard output from the mappers, we can begin development of the ingestion manager. Parallel to this will be the development of a report writing system that is capable of extracting data from the DSpace repository and providing managerial reports. The final development step will be to customize the NRF's new DSpace website to fit the branding of the NRF. All the tools and the DSpace UI will be written in Java or Python, with three different interfaces; one for the automatic and manual mapper, one for the ingestion manager and one for the report writer.

[2] @mire is a registered partner of DSpace - http://atmire.com

We will require the input for the automatic and manual mappers to be in a standardised format (e.g. CSV files) as they are commonly used and many databases allow exporting to CSV files. For the report writing tool we will try to ensure that is remains compatible with the different database management systems DSpace allows.

3.2 Development Methods and Practices

Throughout the course of this project, we will use Agile software development methods. We believe that the requirements and solutions will evolve as we progress and as such, we need to be ready to respond to change easily and early. Our focus will be on evolutionary, iterative development. We will aim to release a minimum viable product early, and through feedback from Hussein and the NRF, we hope to continuously improve on and add features to the project.

Iterative development will be used to break down the project into smaller, more manageable sub-parts. In each iteration, we will plan, develop, code and test the new feature. Each iteration will add onto the main application until we are finally ready to deploy a fully functional piece of software. We will aim to time-box each iteration/sprint into one or two weeks. After each iteration, we can then determine what feature we should develop next that would add the most value to the product.

Depending on the requirements of a specific feature, we will either develop it collaboratively using Bitbucket[3] and Git as a version control system, or independently. Before code is pushed to the master branch, thorough testing will be undertaken to ensure that the master branch is always in a working state. Any bugs will be tracked via the bug tracker on Bitbucket. Unit testing as well as test driven development will be conducted to ensure that code is functioning correctly. In some cases, we will write the tests collaboratively before writing the code as this will help us get a better understanding of the requirements of a specific feature.

Work allocation among team members is discussed in Section 7.6, however this may change slightly as the project scope changes during development.

We hope to meet with our project supervisor every two weeks and will be in frequent e-mail and telephone contact with the NRF. We will aim to have a weekly meeting with each other in which we discuss what we have accomplished in the last week, what difficulties we have had and foresee, what we plan to achieve in the next week and any future work allocations.

User-centered design (UCD) will be used as the process to develop the required user interfaces. Here we will focus on developing the interface based on how it will be used and understood by the end-users of the system. As such, we will focus on ensuring that the interface and system supports the users' requirements and needs. This should ensure that the transfer to the new system is smooth and that the tools we develop are more efficient and user-friendly than the prior system.

3.3 Evaluation Measures and Acceptance Testing

Our system will be evaluated using customer satisfaction measures and possibly usability and acceptance testing. To evaluate the success of the NRF database migration we will ask the NRF to evaluate their satisfaction with the work we have done for them. The NRF will likely evaluate our system compared to their legacy system and the project will be determined a success if our system provided substantial improvements on the effectiveness of ingesting new records into their databases. The measure of how generic our tools are can also be done with comparisons against any similar tools already available. We can also test the tool on a variety of different input sources.

[3] https://bitbucket.org

The evaluation of the additional tools we develop will likely be done through usability tests. We want the tools to be as generic as possible so that they can be used by a larger group of people. We will be engaging with the NRF and possibly other users to provide feedback on the design of software we intend to produce.

4. ETHICAL, PROFESSIONAL AND LEGAL ISSUES

Any experiments conducted for this project would likely be simple usability tests to see if the tools we develop are easy to use and effective. As such, there are likely no major ethical issues that will arise in this regard. As we are in charge of a database migration, we have an ethical responsibility to ensure that we do not intentionally, or unintentionally, modify, delete or add any data to the repository, without the consent of the NRF.

As we are providing a service to a client, we have a responsibility to ensure that we behave in a professional manner and ensure that all correspondence and work is of a professional and high standard.

It is likely that we would make all the outputs of this project open source to allow the community to expand and improve on our work.

5. RELATED WORK

Various projects have been undertaken by others to facilitate the management of research and scholarly output. Scripts have also been written by others to facilitate the setup of DSpace for specific use cases. Some related work is discussed below.

In South Africa, a National Electronic Theses and Dissertations (ETD) portal was developed to provide access to a collection of country specific ETDs as well as to assist in the development and management of ETD programmes at various universities [Webley et al. 2011]. Our project would similarly aim to facilitate access to research, by investigating whether or not we can transform DSpace into a research and information management system.

According to the DSpace website [DSpace 2015], DSpace users are continually adding new functionality to suit their organizations' needs. Many developers and organizations share their work with the community and a wiki[4] has been developed to identify and track various extensions and add-ons that have been developed for DSpace. @mire is a registered service provider for DSpace which offers a number of paid add-ons, including a report writing add-on. This add-on automates the creation of reports, based on information stored in the repository. It allows users to select what metadata to include in the report, develop custom report styles and output the report into multiple formats [atmire 2015].

One of the issues that we are trying to address is to make the initial setup and transfer of data from a legacy system into DSpace more efficient and generic. Many projects have been implemented that make use of scripts to automate the process of creating the archive directory to assist in batch uploading of resources into a DSpace repository. There is a considerable amount of literature documenting the methods used for batch ingestion to populate institutional repositories. Mishra et al. [2007] and Mundle [2007] developed Perl scripts to create the DSpace archive directory for batch imports of ETDs whereas Brownlee [2009] made use of Python scripts to process CSV files (created using Filemaker[5]). Walsh [2010] describes using Perl scripts to migrate data from spreadsheets and CSV files into the DSpace archive format for the Ohio State University's IR. Ribaric [2009] describes the use of PHP utilities for the automatic preparation of ETDs (from the Internet Archive[6])

[4] https://wiki.duraspace.org/display/DSPACE/Extensions+and+Addons+Work
[5] http://www.filemaker.com
[6] http://www.archive.org

for deposit into DSpace. Several other projects are introduced by Walsh [2010]. What this shows is that a large number of organisations are writing highly specialised, single use, scripts to initialize and populate the archive. Thus there is a clear need for a generic tool that can be used in numerous circumstances.

6. ANTICIPATED OUTCOMES

6.1 System

As previously stated, this project is mainly a Software Engineering project. We will be developing software tools for DSpace, but we will also customise the DSpace UI for the NRF. We will customise the JSP Web user interface bundled with the DSpace repository open source code. The Web UI will be customised to match the branding of the NRF. We will also install and configure the DSpace repository system on the servers at the NRF and migrate the data from their legacy database systems to the DSpace repository.

The tools developed as part of this project will include an automatic mapper, a manual mapper, an ingestion manager and a report writer. The automatic mapper will allow a user to input a CSV file and will automatically attempt to map the fields of this file to the appropriate DSpace metadata fields. This mapping will be done using methods such as regular expression matching and dictionary lookups. The manual field mapper will allow a user to manually map a field to the applicable DSpace metadata field. This function will be used to correct errors that the automatic mapper may make and set the field descriptor for the fields that the automatic mapper cannot determine. The manual and automatic mappers will have the same Web-based graphical user interface.

The ingestion manager will take the mapping generated by the automatic and manual mappers and generate a file structure and the accompanying metadata files. The ingestion manager will also interface with the DSpace command line ingestion tool to import the data into the repository. It will have its own Web-based graphical user interface and will stand alone from the manual and automatic mappers. The ingestion manager will consist of a file structure builder which will take in a standard mapping format, which is that output of the manual and automatic mappers, and generate the DSpace required file structure and files for ingesting into the repository. It will also consist of a terminal runner that interfaces with the DSpace terminal ingestion tool to ingest the files into the repository. Figure 1 shows the components of the mappers and ingestion manager.

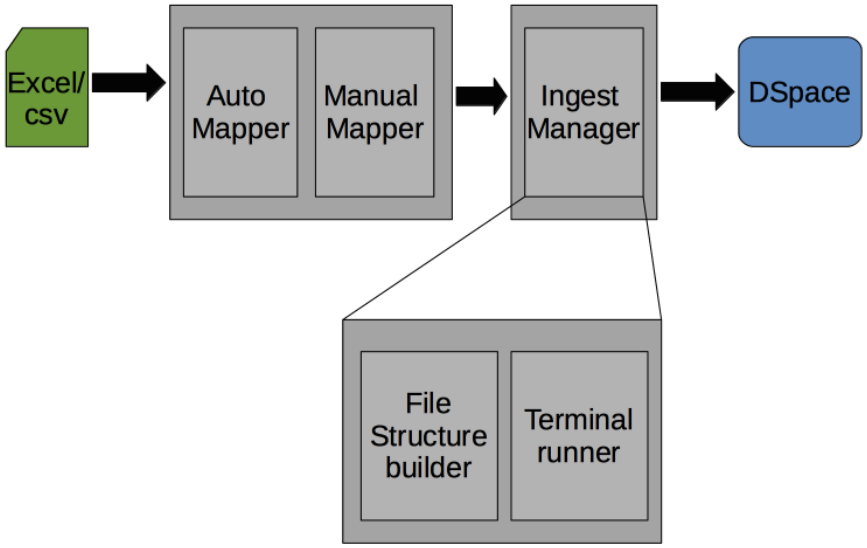


Figure 1: Mappers and ingestion manager components

The report writing tool will read information from the repository and allow the user to generate managerial reports. The reports they produce will be user customisable and allow the user to select which metadata fields to display. The report writer will have its own JSP and Java Servlet Web user interface. It will likely incorporate an open source Java report writing tool that we can modify to integrate with the DSpace repository. The open source reporting tool we are proposing to use is OpenReports[7], which uses the open source JasperReports[8] reporting engine. Figure 2 shows what the components of the report writer are likely to be.

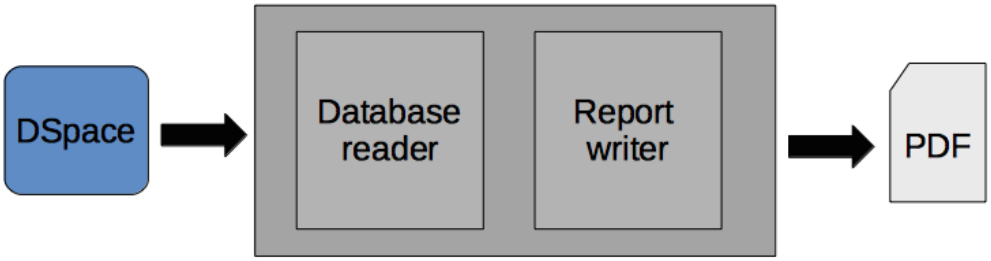


Figure 2: Report writer components

In Figure 3 below the core features of the RIMS software InfoEd[9] are mapped against similar available feature in DSpace. This graphic shows the gaps that we have identified in the DSpace system that we hope to fill by creating these tools. The tools we will be creating are highlighted in blue. We hope that these tools will help transform DSpace into an RIMS.

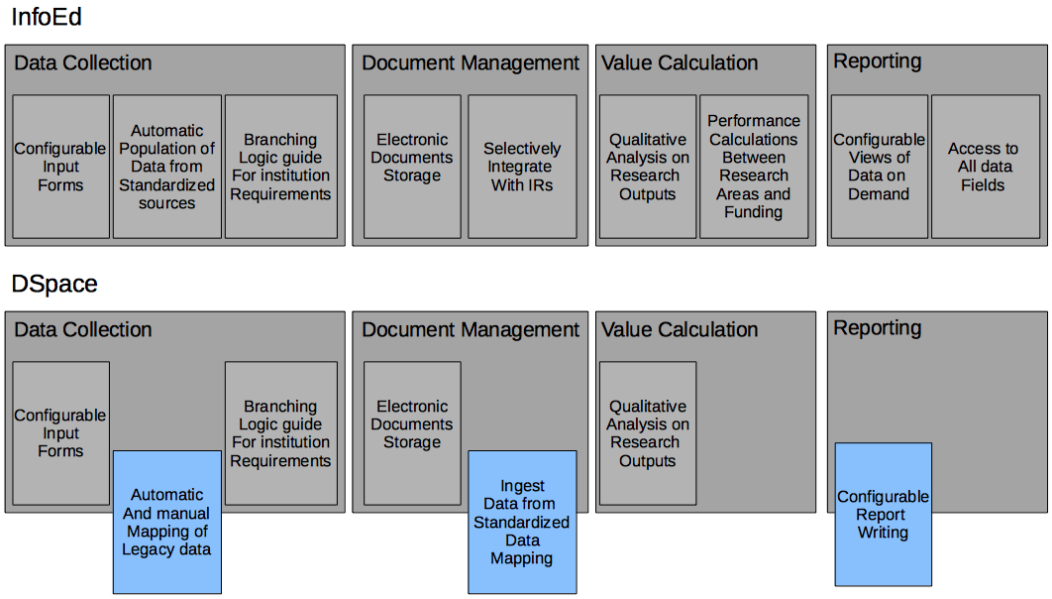


Figure 3: Core features of InfoEd mapped against similar available features in DSpace

[7] http://oreports.com/
[8] http://community.jaspersoft.com/project/jasperreports-library
[9] http://infoedglobal.com/solutions/research-outputs

6.2 Expected Project Impact

We hope to produce a project that will have a positive impact on the current DSpace and RIMS communities as well as users looking to migrate their systems to DSpace. By releasing the tools as open source, the community will be able to freely use our work and expand and improve on our tools.

The automatic metadata mapping along with the generic importing tool should help to reduce the time required to add items into an existing or new DSpace repository. The NRF has specifically requested that their legacy system be migrated to DSpace as this will help reduce the time spent by the NRF capturing data. Universities will be able to submit the required information directly to the DSpace repository and the NRF would merely be required to approve this addition or modification.

It is clear that this project should have a meaningful, positive impact at the NRF and hopefully in the broader community as well.

6.3 Key Success Factors

The key success factors for NRF database migration will be:

- The new DSpace system must correctly display all the information that the legacy database system did without errors or omissions.
- The new DSpace system must allow users from different universities across the country to submit new records into the system. The NRF must be able to validate those submissions before they are seen by the public and that the submission, if validated, must be viewable by the public.

The key success factors for the automatic and manual report mapper tool will be:

- The ability to successfully map certain fields (possibly, but not limited to, author(s), date, title, abstract) with a high probability. Due to the variability of the legacy data from various sources, it is not possible to say that the automatic mapper will be able to map every field correctly, all the time.
- The manual report mapper must be able to correct any mapping errors or omissions the automatic mapper makes, by allowing the user to select what the field name is from a predefined list of names.

The key success factors for the ingestion manager tool will be:

- The ingestion manager must be able to take in the mapping that the automatic and manual mapper generated and output the correct file structure and files that DSpace expects.
- The ingestion manager must also be able to take the file structure and files it creates, together with the necessary input from the user, and import the data into the DSpace repository.

The key success factors for the report writing tool will be:

- The report writer must be able to read information from the DSpace repository and allow the user to compile custom reports based on that information.
- The reports must present the user with meaningful information about the content of the repository.
- The report writer must allow the user to export the report to a standard file format such as PDF.
- The report writer must produce formatted, customisable reports.

7. PROJECT PLAN

7.1 Risks and Risk Management Strategies

There are a number of potential risks related to this project. The major risks, coupled with their mitigation, monitoring and contingency plans are attached in Appendix A.

7.2 Timeline

The Gantt chart for this project is attached under Appendix B.

7.3 Required Resources

In order to test the tools we develop, we would require test data. It would be necessary to have official data from the current NRF system to ensure that the system will work with the current NRF system. Other test data from different sources would prove useful in verifying and testing that the solution we develop is generic enough to be applied to other systems. We may need to be in contact with the I.T. department of the NRF to get the required data.

We would need to have access to a local or remote version of DSpace, however, this should not be an issue as the software is free and open source.

Initially, we would make use of our local machines to run the DSpace server, however, as the project progresses, we hope to make use of one of UCT's servers until the project is complete. Setting up the DSpace repository on a UCT server would also provide us with valuable experience which should facilitate the final implementation at the NRF.

7.4 Deliverables

The deliverables for this project are listed below. The order they are listed in is not necessarily the order they will be delivered. For specific dates see the Gantt chart attached under Appendix B.

- A presentation of the project proposal.
- A project Web presence including the project proposal and timeline.
- An automatic mapper that allows a user to input a CSV file and will attempt to automatically map the fields to the appropriate DSpace metadata fields.
- A manual field mapper that allows a user to set the field descriptor from a list of predefined names. This will be used to correct errors and set the field descriptor for the fields the automatic mapper cannot determine.
- An ingestion manager that takes the mapping from the automatic and manual mapper and generates a file structure and the accompanying files for the legacy data. The ingestion manager will also interface with the DSpace command line ingestion tool to import the data into the repository.
- A report writing tool that reads information from the repository and allows the user to generate managerial reports.
- A customised DSpace JSP user interface with NRF branding.
- Installation and configuration of a new DSpace instance onto the servers at the NRF.
- A demonstration of the initial prototype.
- The background/theory section of the final project report.
- The design section of the final project report.
- A first implementation with performance testing and write-up.
- A final prototype with performance testing and write-up.
- An outline of the final project report.
- A complete draft of the final project report.
- The final project report.
- A project poster.
- A website for the project.
- A reflection paper based the project.

7.5 Milestones

For a list of the milestones for this project see the Gantt chart attached under Appendix B where milestones are specified as tasks.

7.6 Work Allocation

The project consists of two major sections. The two major sections are report generation and the automatic/manual mapper tool to add the data into DSpace. The work would be divided as follows: Craig would work on the automatic and manual mapper and Darryl would work on ingestion and report writing. All other tasks will be shared equally between the team members.

REFERENCES

ROAR. 2015. Registry of Open Access Repositories. Retrieved May 10 2015 from http://roar.eprints.org/.

Lawrence Webley, Tatenda Chipeperekwa and Hussein Suleman. 2011. Creating a National Electronic Thesis and Dissertation Portal in South Africa. In *Proceedings of the 14th International Symposium on Electronic Theses and Dissertations (ETD2011)*, Cape Town, South Africa, National Research Foundation.

atmire. 2015. Listings and Reports | atmire. Retrieved May 10 2015 from http://atmire.com/website/?q=modules/lr.

DSpace. 2015. Add-ons and Extensions | DSpace. Retrieved May 10 2015 from http://www.dspace.org/addons.

R. Mishra, S. Vijayanand, KPP Noufal and Gaurav Shukla. 2007. Development of ETD Repository at IITK Library using DSpace. In *International Conference on Semantic Web and Digital Libraries (ICSD-2007)*, Bangalore, India, Indian Statistical Institute, 249-259.

Todd Mundle. 2007. Digital retrospective conversion of theses and dissertations: an in house project. In *8th International symposium on electronic theses and dissertations (ETD 2005)*, Sydney, Australia, NDLTD.

Rowan Brownlee. 2009. Research data and repository metadata: policy and technical issues at the University of Sydney Library. *Cataloging & Classification Quarterly* 47, 3/4, 370-379.

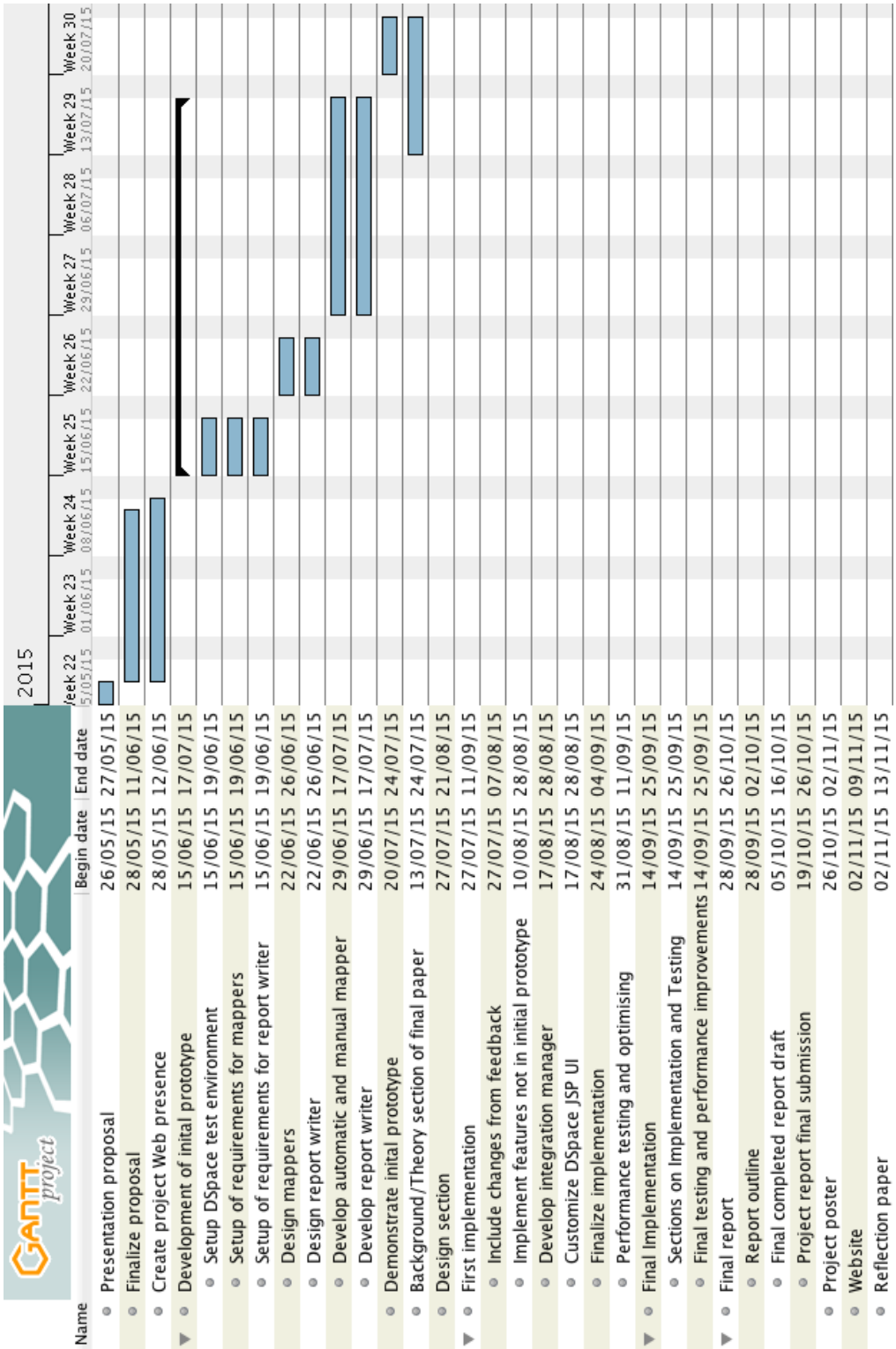
Maureen P. Walsh. 2010. Batch Loading Collections into DSpace: Using Perl Scripts for Automation and Quality Control. *Information Technology and Libraries* 29, 3, 117-127.

Tim Ribaric. 2009. Automatic Preparation of ETD Material from the Internet Archive for the DSpace Repository Platform. *Code4Lib Journal* 8.

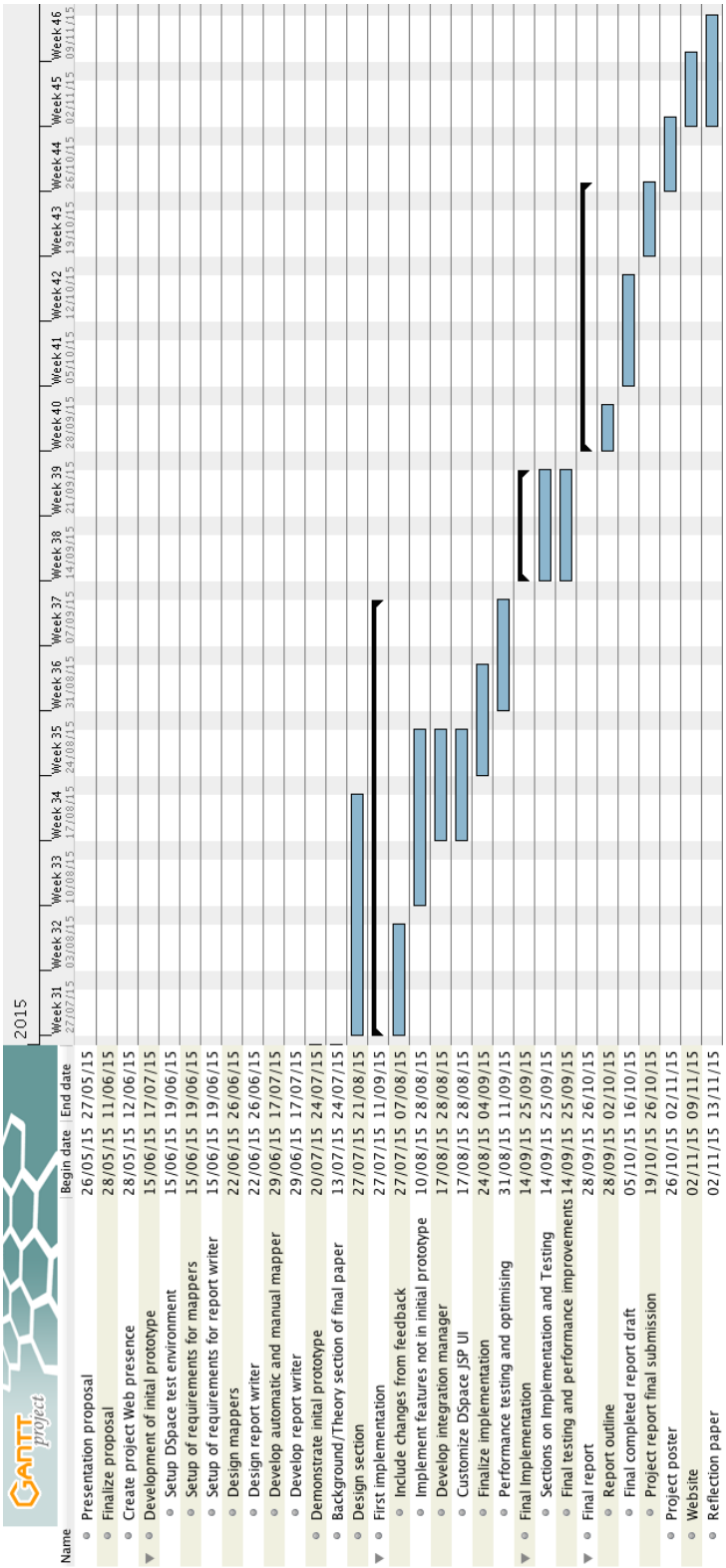
Appendix A – Risk Management Table

No.	Risk Condition	Consequence	Cat	Prob	Impact	Mitigation	Monitoring	Management
1	Requirements specification inflating to increase scope.	Lead to the time required for development increasing and it would shift the focus from the more important features.	Development	Low	Critical	Ensure requirements are well defined and understood prior to development.	Communicate with client to make them aware of current progress so that new features are not added during development.	Drop non-essential features and budget for more development time within the project timeframe.
2	Underestimating the time required to complete each phase of the project.	Fall behind schedule and be forced to submit sub-standard work.	Planning	Medium	Critical	Ensure accurate estimation for required time for each stage during planning phase.	Keeping track of progress, ensuring deadlines are met and making adjustments if necessary.	Understand what went wrong to prevent further occurrences.
3	Temporary absence of a team member.	Falling behind schedule and increased workload on other team members.	Length of project	Medium	Marginal	Communication between team members to make them aware of future temporary absences and the need to redistribute workload accordingly.	Constant communication between team members.	Workload is redistributed to ensure the team does not fall behind schedule.
4	Permanent absence of a team member.	Falling behind schedule and increased workload on other team members.	Length of project	Low	Critical (depending on departure time)	Communication between team members to make them aware of a possible absence and the need to redistribute the workload accordingly.	Constant communication between team members.	Workload is redistributed to ensure the team does not fall behind schedule. In the worst case scenario the scope has to be adjusted.
5	Conflict between team members.	Communication breakdown could lead to the project falling behind schedule.	Length of project	Low	Marginal (depending on severity of conflict)	Respect the contribution of other team members and allow for open communication between team members.	Observe the way that team members communicate with one another to identify possible future conflicts.	Introduce a neutral third party to help mediate and resolve the conflict.
6	Functional specifications are not met and the software is incomplete at the end of development.	Incomplete software will be submitted to the client.	Planning	Low	Catastrophic	Ensure the project remains on track during development and unnecessary features that would lead to delays are not added.	Keep track of progress and compare current progress against projected progress.	Renegotiate deadline with client to allow for extra time to complete project (not feasible for Capstone project).
7	Adding unnecessary features before completing the core features.	Project delays and non-complete final product.	Planning and Development	Medium	Marginal	Focus on implementing core features first and understand what features are core and non-core.	Ensure all team members give priority to developing the core functionality until complete.	Return to focusing on the priority code as soon as possible.

Appendix B – Gantt Chart (1/2)



Appendix B – Gantt Chart (2/2)



Name	Begin date	End date
Presentation proposal	26/05/15	27/05/15
Finalize proposal	28/05/15	11/06/15
Create project Web presence	28/05/15	12/06/15
Development of inital prototype	15/06/15	17/07/15
Setup DSpace test environment	15/06/15	19/06/15
Setup of requirements for mappers	15/06/15	19/06/15
Setup of requirements for report writer	15/06/15	19/06/15
Design mappers	22/06/15	26/06/15
Design report writer	22/06/15	26/06/15
Develop automatic and manual mapper	29/06/15	17/07/15
Develop report writer	29/06/15	17/07/15
Demonstrate inital prototype	20/07/15	24/07/15
Background/Theory section of final paper	13/07/15	24/07/15
Design section	27/07/15	21/08/15
First implementation	27/07/15	11/09/15
Include changes from feedback	27/07/15	07/08/15
Implement features not in initial prototype	10/08/15	28/08/15
Develop integration manager	17/08/15	28/08/15
Customize DSpace JSP UI	17/08/15	28/08/15
Finalize implementation	24/08/15	04/09/15
Performance testing and optimising	31/08/15	11/09/15
Final Implementation	14/09/15	25/09/15
Sections on Implementation and Testing	14/09/15	25/09/15
Final testing and performance improvements	14/09/15	25/09/15
Final report	28/09/15	26/10/15
Report outline	28/09/15	02/10/15
Final completed report draft	05/10/15	16/10/15
Project report final submission	19/10/15	26/10/15
Project poster	26/10/15	02/11/15
Website	02/11/15	09/11/15
Reflection paper	02/11/15	13/11/15