Develop an RDF store for Phenotype data

Report Name	Progress Report
Author (User Id)	Adam Connah (aoc9)
Supervisor (User Id)	Georgios Gkoutos (geg18)
Module	CS39440
Date	November 18, 2012
Revision	1
Status	Release
Word Count	3626

Adam Connah (aoc9)

Contents

1 Project Summary

1.1 Brief Introduction of Project

The project I am undertaking for my final year is to create an RDF store that can be used to provide a fast analysis of the similarity between different phenotypes in animals. This knowledge can then be used to aid our understanding of the function of genes, by systematically analyzing the phenotypic outcome of their mutation and ultimately facilitate our ability to prioritize causative genes for rare and orphan diseases by enabling the comparison of the phenotypic similarities between experimental data and human clinical signs and symptoms.

The system will be made available via a web-based platform, so it can be accessed from anywhere with an Internet connection, and there are no minimum specifications in terms of hardware or software, except for a web browser. The web interface is an imperative part of the project, as it provides the tools for a user of any technical ability the opportunity to search the vast amounts of data, by using simple keywords. This interface forms an essential part of the project since it will allow users to query and analyse the data without requiring knowledge of a query language.

This interface will allow queries to be ran on the server, and retrieve the data It will have to present itself in such a way that the user can easily understand what they have been given. Again this ties in with the ease of use, and the lack of any technical knowledge on the user's behalf.

In the current day and age, the field of Biology has vast amounts of data that they are unable to quantify quickly and, as a result, they need a system that will enable them to do so. This is what the project hopes to achieve, and if successful it will directly influence and aid biologists.

The project has a huge potential in terms of the impact it can have, for good, on the world that we live in. Using the application biologists will be able to draw upon a huge resource of data, that will be constantly expanding, from all across the globe. The prospect of playing a small part in the discovery of a new understanding or knowledge of a disease or illness, which could lead to providing a better quality of life to people, is a fantastic opportunity. This project has the potential to become a very powerful tool, and one that can be used out in the world for a great purpose.

Combined with the aspects mentioned above, is the chance to learn new technologies and how they integrate with each other. I had no prior knowledge of Resource Description Framework (RDF) before accepting this project, and the same applies with SPARQL, as well as gaining a greater understanding of PHP. I feel this will interest me a lot as I have a desire to develop my skills relating to Web technologies.

The project will require a combination of PHP, and XHTML to create the web interface, which will be integrated with the back-end of the system, to produce the final project.

1.2 Project goals

The aim of the project is to produce a web application with an interface that allows a user with very limited computer knowledge to query the RDF store. There will be a need for a Triple Store (RDF database), using a suitable Data management system. This will mean that there needs to be an interaction between the web interface and the Data Management System, and to do this PHP will be used. The project will also require the transformation of existing SQL data to RDF/XML format so it can be read into the application.

1.3 Project desirables

Some desirable aspects of the application would be the ability to include a comparison between different diseases. This would be done using the web application interface, whereby a user would select a disease from the search results they get, and view a side by side illustration of the different phenotypes. This builds upon the idea of having a "similarity value", as seen in the PhenomeNet project, and allows a quick and efficient visual representation, which the user can use to help their studies. This could also be done the other way round, by comparing phenotypes and producing a similar representation of the diseases that are related to the phenotypes. This idea will also open up the possibility of a tree structure, where the user manually searches through the different Phenotypes and compare them.

1.4 Possible Project Limitations

The limitations of the project center around the user and what they wish to achieve. It might be possible that in a user wants to use more complex query techniques, and in this case, they would have to either learn how to use the SPARQL query language, or they could be provided with a set of preformed query examples to use.

2 Current Progress

2.1 Related Works

There are a few projects that are similar to this one, from a biological point of view, but are using different technologies. For example, PhenomeNet, is a "*Cross Species Phenotype Network"*, which uses MySQL to produce a very similar system.

RDF is most widely used in Academic projects at the moment, and an example of this is http://bio2rdf.org/. Similar to PhenomeNet this is used in Bio-informatics, and their aims, taken from the website mentioned, is to "*Create machine interpretable data that can be powerfully interrogated with SPARQL-based queries to answer sophisticated questions".*

PhenomeNet, is probably the most closely linked to the project as it aims to produce a very similar design for the web application, in terms of simplicity and functionality. It is also a good idea to keep it similar so that users do not get confused when switching between the two.

One of the most pleasing aspects about PhenomeNet, is that it is easy, for the user, to realise where to begin. The homepage provides a quick explanation of the project itself, and what it can be used for, and below it has a search box. There is no unnecessary

clutter, around the page, and this is something the interface of my project would like to try and emulate. PhenomeNet also has a help page, which explains which key words would be suitable to use, and this will help the user to optimise their search phrases to gain better sets of results.

A key feature of PhenomeNet is the 'similarity value', which indicates how similar two diseases are to each other in terms of their symptoms. The project will look to build upon this idea, and introduce a 'compare' feature, as I have discussed about in section 1.3 of this report.

Another example of related work, is Bio2RDF, which this time uses RDF to provide a huge resource that can be queried by the user. However, it can be confusing at first to understand what is being required from the user. Additionally the way the data is represented is not very well structured when you do get results. This could be a cause of concern with RDF, and the project will need to find a way to ensure that it is represented in an understandable manner.

2.2 Relevant Technologies and Related Literature

As mentioned above, I had limited knowledge of many of the technologies needed to produce this project before starting, and as a result have had to do a lot of reading to understand the capabilities of them all.

To start with there was a need to select a suitable Data Management System. After researching relevant applications, Virtuoso was chosen as, the most suitable for the projects demands.

The project also uses two fairly new technologies and languages, namely RDF and SPARQL. A related piece of literature is the “Introduction on RDF”, (http://www.rdfabout.com/quickintro.xpd), and this will help to develop an understanding of the uses of RDF, and why it is helpful in this project, compared with other technology such as XML.

A book on RDF, available from the Physical Sciences Library at Aberystwyth University, titled “*Practical RDF – O'Reilly”*, will be helpful throughout the whole project, as it covers basic to advanced features, and will be a good place to go for help.

As mentioned, Virtuoso is the chosen Data Management System for the project, and there will be a need to assess the abilities of the application. Virtuoso is open source project, and as a result has lots of documentation and tutorials on their website which are very nicely written. This was very helpful when installing Virtuoso onto a Linux machine, so that the project could be tested locally. These tutorials also discuss the command line interface, ISQL, which allows you to enter SPARQL statements to upload and query the RDF data.

The Yahoo User Interface (YUI) JavaScript Library will aid my development of the web interface, as it is designed to work with applications that are using AJAX like techniques, and this project will operate as a 'RDF-AJAX' interaction.

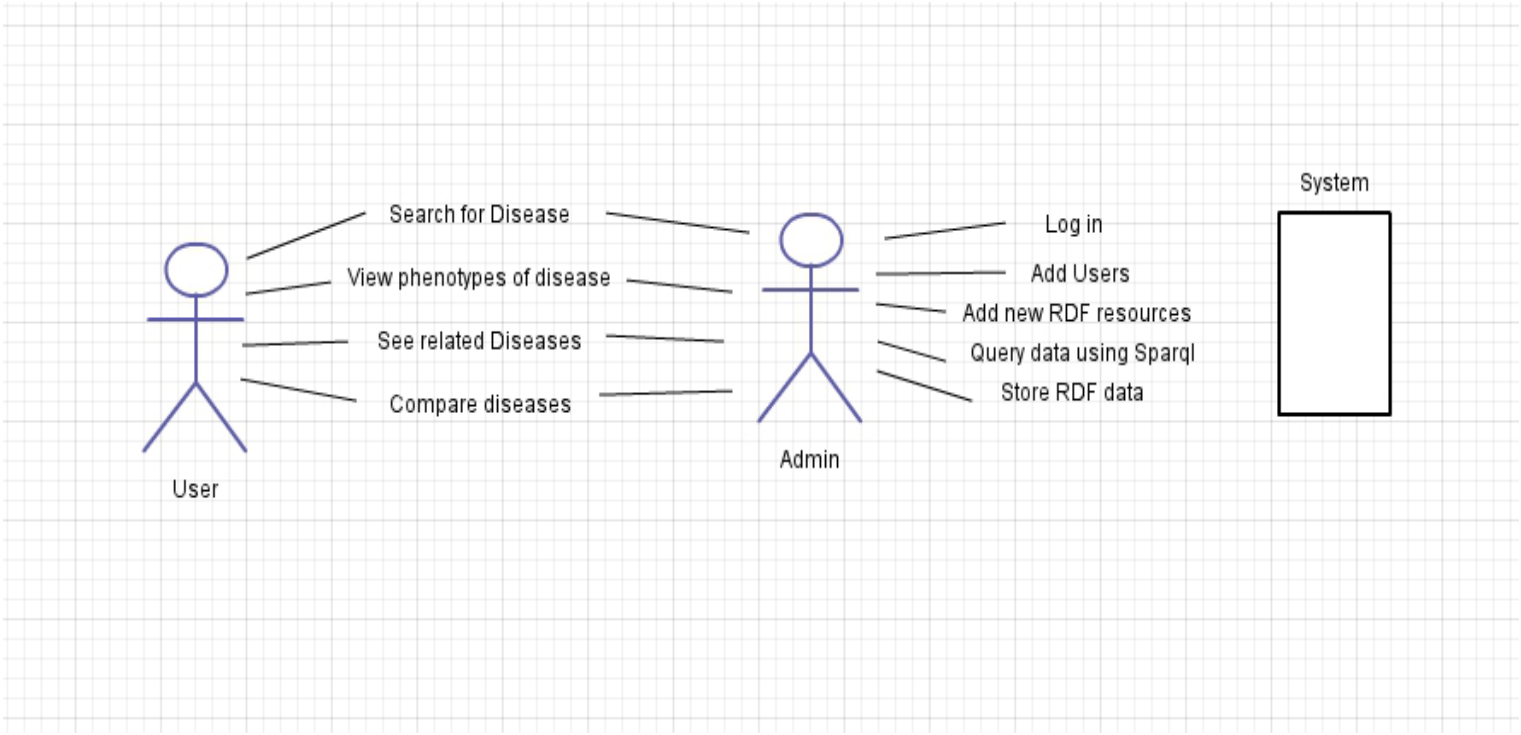
2.3 Prototypes and New Technologies

In terms of Prototypes, I am aiming to use small data samples to test the capabilities and functionality of Virtuoso's features. It is a good idea to do this, instead of waiting until the XML data has been converted into RDF format, at which point the data sets would be a lot bigger, and so the process is more vulnerable to errors.

The results of this process could vary, for instance if Virtuoso wasn't able to handle the RDF format, then the whole process would have been a waste of time, so it is important to establish that it is working correctly with some small and basic RDF files. It will also provide an understanding of the limitations that the technologies may have, which can be catered for from an early stage. This process will not just focus on the negatives however, as it will also help to understand the positives aspects of Virtuoso, and what it can do well.

For the interface design, a few mock-up designs of the web application have been constructed, but they still require a bit more work before they are completed. These include the basic layout of pages, such as Homepage, About, and other similar pages, as well as deciding how the search results will be presented. These will provide a basis to build upon, and this will allow for a better understanding of the project, without setting up the whole system, and it will also provide any insight into flaws in the layout, or anything that may have been overlooked. It is a lot simpler to iron out any problems when the design is only a mock-up, and would be much harder to change once the system was fully set up.

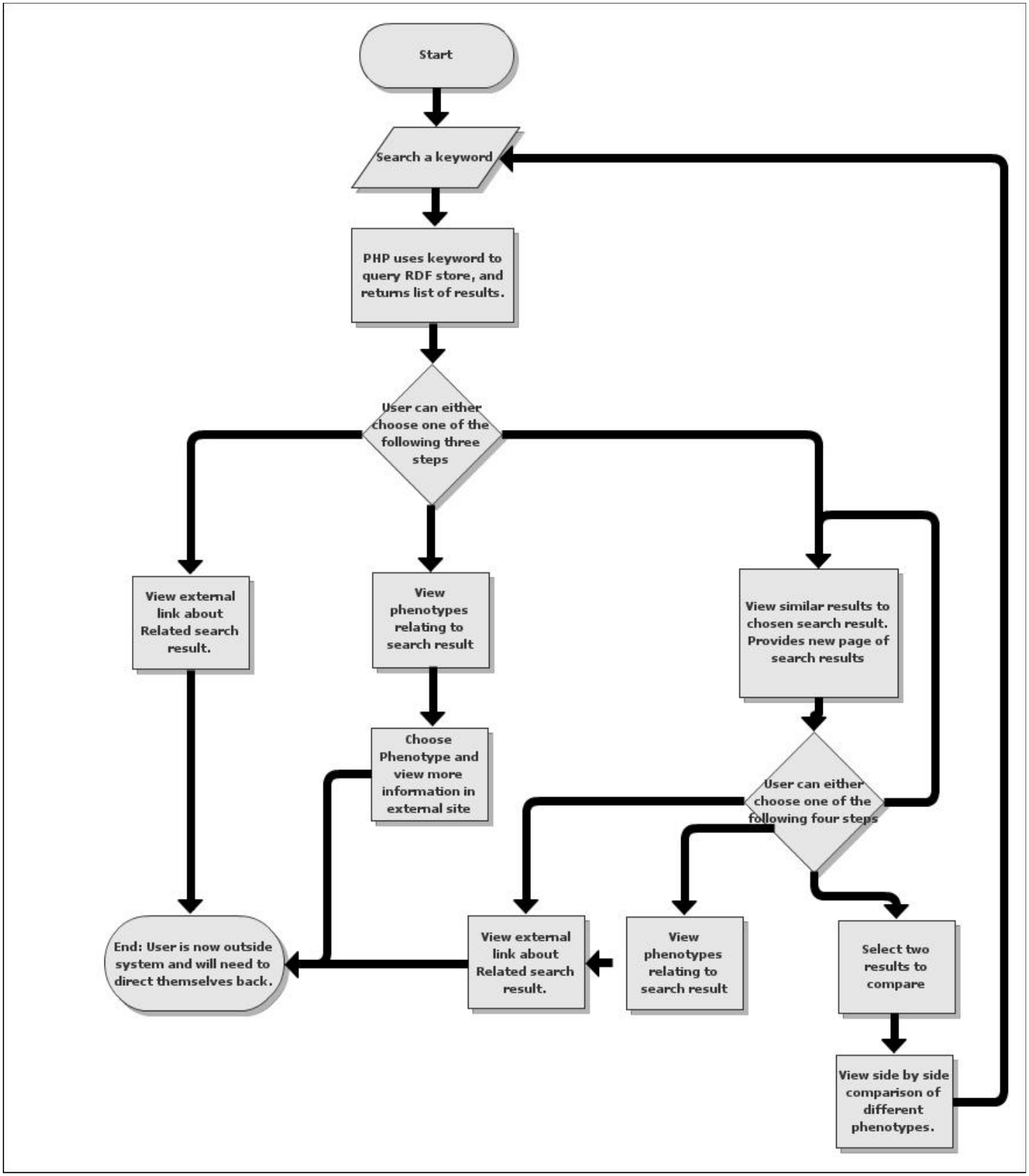
2.4 Outline of Design



The above Use Case diagram, displays the functional requirements of the system, for the User and Administrator. The system is not so complex, as the Users do not have to log in to anything, or leave any details, this means their visits will anonymous as no data will be stored of it.

It gets a bit more complex for the Administrator, as they will need to be able to log into the system to possible perform maintenance, or update the data. One of the key features of RDF, is that there isn't a need to constantly update database files. However there is still a need to add new resources, when they become available and this will be managed through the Virtuoso Universal Server.

The system will allow for all of these events to occur, and so it is in constant communication with the user. This is why it is represented above on the diagram, as all of the actions will have to go through the system.



Above is a flow diagram that shows the route a potential user may take through the system. As shown, the route for the user is very straightforward, and it allows the user to keep looping through similar sections of the website before settling on the result they require. What this diagram doesn't account for is that each page will have a search box, so this means at any stage the user can decide to begin searching for something else, and this takes them back to the start of their route.

The most notable thing, that can be taken from this diagram, is that the user often ends up outside of the website. This shouldn't be too much of a problem, as most users are accustomed to using the back button on their browsers, and the main aim of the application is to provide users with the information they require. It may lead to users becoming a bit frustrated, so perhaps this is one of the small disadvantages with RDF.

2.5 Customer and Target User

The project doesn't have a specific customer, instead there is a target audience that the application will cater for. This mostly relates to the Web interface, as the user would never have any dealings with the back-end aspect of the application.

In terms of technical ability among users, this could be very vast, and so the project will need to ensure that it can be used by people with even the most basic of computer skills, whilst the same time, it needs to be made in such a way that a skilled user does not find it frustrating. As an example, Google.co.uk, is one of the most easy to use websites on the Internet due to its simplistic homepage. It is obvious for a user that all they need to do, is type in a keyword, and they will be provided with a list of results, and this is the type of user experience the project should aim to emulate,

In terms of area of expertise, the application would mostly be used by people who have an understanding of Biology or a related field. The application would not be of used to anybody who does not share this expertise, and so the terminology that can be used throughout the website can be aimed at this audience.

Our goal is to create a project that embodies both of these aspects, and if it does, it will be able to provide the experience that the user requires.

2.6 Explanation of Technical Environment and Toolkit choices of Key Aspects

Below is an explanation of why the choices of technologies and languages are necessary for the project to be successful.

2.6.1 RDF

Using an RDF store is key to the project as it is a decentralized system, and hence it facilitates the retrieval and analysis of data that resides in a variety and diverse sources. Such a store will generate a large resource that can be accessed quickly and efficiently providing biomedical researchers and bioinformatics alike a valuable resource and tools for their analysis. One of the biggest impacts of RDF is that it is designed to be read and understood by humans, and not for a machine, unlike XML.

2.6.2 Virtuoso Universal Server

Adam Connah (aoc9)

Virtuoso is a very powerful application in its own right, allowing for command line, or browser interaction. It incorporates a vast amount of different aspects of common database functions, such as Content management and even Mail storage, as well as other abilities, into one place thus the “universal” part of the name. The main features related to the project, is it's capabilities with being able to hold an RDF store, as well as being compatible with the SPARQL query language.

2.6.3 SPARQL

This is a similar concept to languages like MySql, which allows users to query databases. In this instance the SPARQL language will be used to query the RDF store. This was a straightforward choice, as SPARQL is a W3C Recommendation and so it has the full support of the World wide web consortium. It also seems SPARQL has similarities to query languages such as MySQL, as well as being compatible with Virtuoso, and this makes it a prime choice for the query language.

2.6.4 YUI

In regards to the web interface, there are a number of different toolkits that would be suitable to the project. YUILibrary was the chosen one, for its huge library, and more specifically, YUI PHP Loader Utility. Googleweb toolkit was a possible choice, but it seems to have a lack of compatibility with PHP. Additionally, the YUI PHP Loader, has huge amounts of documentation, and because of its Open source status, it is constantly being updated with new features from its community.

The PHP Loader utility provides the project with the basis to interact with the data stored on the Virtuoso server, with the front end web browser. There will need to be more research into this when it is time to implement the feature, but I am confident enough after reading the documentation that comes with it, that it will provide everything that the project needs.

3 Planning

3.1 Chosen Methodology

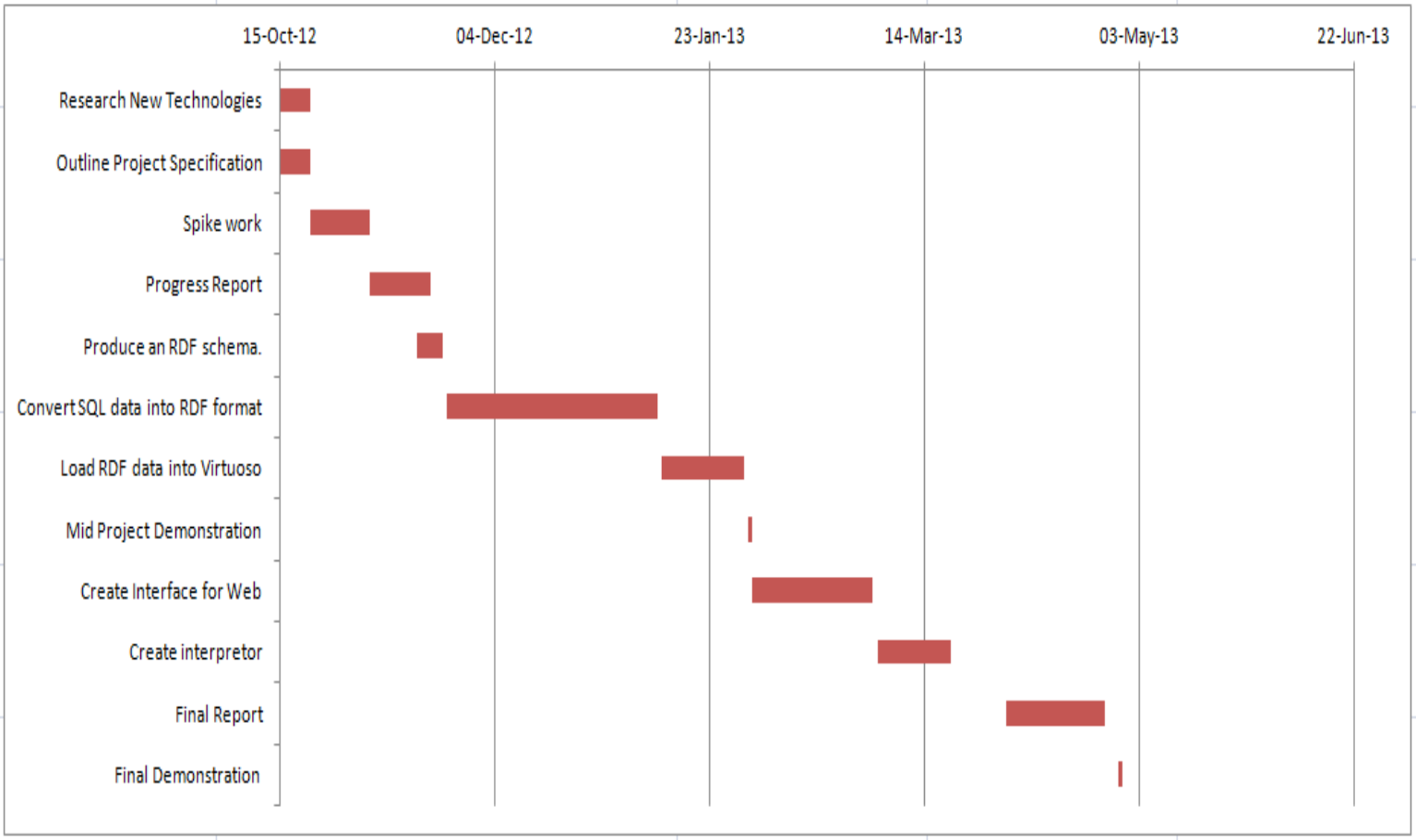
For the project, the Waterfall model will be a very solid life-cycle to adhere to during development. One of the key requirements for the Waterfall model to be successful, is sufficient planning in the early stages of production, as well as requirement specifications that are not subject to change. In this sense it is quite rigid, but as the project has a specific goal in mind, the requirement specifications are not going to alter, and this approach suits the project well.

The other aspect of the project, is that it can be approached in stages in regards to the implementation. This means that the linear approach of the Waterfall Model compliments the project, and will work well. After each implementation there is some testing, which has a high value to the project, as it allows any problems to be weeded out early on in production. The documentation approach also goes hand in hand with this, as it is produced along side the production of every stage. This will enable the production of a strong final report, which contains information from every implementation.

The Integration stage of the methodology also relates well to the project, and it will help to make sure the various different parts of the system fit together properly and function together. This will also require testing which is the next stage of the methodology, and it will allow for one final assessment of the system, focusing on functionality and performance.

Finally there is the Maintenance stage, and while this is not required for the project itself, it would be very easy to do in the future. This is due to the features of the Virtuoso universal server and the flexibility of RDF.

3.2 Gantt Chart



Above, is a Gantt Chart, used to represent the different parts of the project during production, and the expected length of time each part will take to implement. These dates are subject to change and if this happens there will be a knock on effect with the rest of the project. To minimize disruption to the project, this graph shows an exaggerated estimate for each part, which will allow for any problems that may occur.

Testing the different aspects was also taken into account when creating this graph, and has been reflected in the chart. For example, the stage named 'Convert SQL data into RDF format' will be expected to be completed in less time than shown above, and the remaining time can be used to test that it has been completed to a high standard.

3.3 Mid-Project demonstration

3.3.1 Working aspects

Adam Connah (aoc9)

The aim for the Mid-Project demonstration is to have the RDF store set up to a degree whereby it could be queried using the SPARQL query tool inside Virtuoso . This would mean that some or all of the data would have been loaded in, and that the data currently stored in XML format, will need to be converted into RDF format to allow it to be uploaded to the system and queried properly.

If this is not possible, then there will be the need to use other example sources, that are already in RDF format, so as to demonstrate how the system would behave. Ultimately though the project aims at demonstrating the functionality of the Virtuoso Universal Server in relation to the needs of the project.

However this would mean that what needs to come next is the creation of the Interface, for the web browser. This would include the interpretor, written in PHP, between the Interface and Virtuoso to allow the queries to be done from the browser, and without being written in SPARQL query code.

3.3.2 Technical Requirements

For the demonstration, the only thing that I would require is a machine inside the University network that I could use to log into Virtuoso in a web browser.

3.4 Final Demonstration

3.4.1 Working Aspects

For the final demonstration, the aim is to have a fully functional web application, capable of receiving keywords and using them to query the data held in the RDF store.

The demonstration will show what the user can achieve through the application. This will involve a demonstration of biological examples that demonstrate how the system can be exploited by biologists and what the advances of exploiting the data can bring in the field. The various different features of the system, will be represented, such as the ability to add new RDF resources, from file or URI. This will demonstrate the flexibility of the application, and how scalable it is in the long term.

There can also be a demonstration of some of the advanced features, for example, using SPARQL to write more complex queries to search the database.

3.4.2 Technical Requirements

As above, I won't need much to run this, except for a machine inside the University network.

4 Annotated Bibliography

Graphite PHP Linked Data Library. Available at: <http://graphite.ecs.soton.ac.uk/> [Accessed on 25th October 2012]

Google Web Toolkit PHP Framework. Available at: <http://www.gwtphp.com/> [Accessed on 25th October 2012]

RAP is a software package for RDF models. Available at: <http://wifo5-03.informatik.uni-mannheim.de/bizer/rdfapi/> [Accessed on 25th October 2012]

Virtuoso Universal Server. Available at: <http://virtuoso.openlinksw.com/> [Accessed on 25th October 2012]

An Introduction to RDF. Available at: <http://www.w3schools.com/rdf/rdf_intro.asp> [Accessed on 25th October 2012]

Creating an RDF Store. Available at: <http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/index.jsp?topic=%2Fcom.ibm.swg.im.dbclient.rdf.doc%2Fdoc%2Fc0060567.html> [Accessed on 26th October 2012]

Ajax and the Semantic Web. Available at: <http://fgiasson.com/blog/index.php/2005/09/26/ajax_and_the_semantic_web/> [Accessed on 26th October 2012]

Quick Introduction to RDF. Available at: <http://www.rdfabout.com/quickintro.xpd> [Accessed on 26th October 2012]

Longer introduction to RDF. Available at: <http://www.rdfabout.com/intro/?section=contents> [Accessed on 28th October 2012]

PhenomeNet – Cross Species Phenotype Network. Available at: <http://phenomebrowser.net/> [Accessed on 28th October 2012]

Introduction to Google Web Toolkit. Available at: <https://sites.google.com/site/angelhurtado/tutorialgwt2> [Accessed on 30th October 2012]

Virtuoso Universal Server Wikipedia page. Available at: <http://en.wikipedia.org/wiki/Virtuoso_Universal_Server> [Accessed on 2nd November 2012]

Yahoo User Interface. Available at: <http://yuilibrary.com/projects/phploader> [Accessed on 2nd November 2012]

Yahoo User Interface Projects. Available at: <http://yuilibrary.com/projects/> [Accessed on 2nd November 2012]

Yahoo User Interface, PHP Loader. Available at: <http://developer.yahoo.com/yui/phploader/> [Accessed on 2nd November 2012]

Introduction to YUI PHP Loader. Available at: <http://www.slideshare.net/chadauld/introduction-to-yui-php-loader> [Accessed on 2nd November 2012]

Using N3 with RDF. Available at: <http://www.w3.org/2000/10/swap/Primer.html> [Accessed on 2nd November 2012]

A Direct Mapping of Relational Data to RDF. Available at: <http://www.w3.org/TR/2011/WD-rdb-direct-mapping-20110324/> [Accessed on 2nd November 2012]

Bio-informatics Organisation's own wikipedia. Available at: <http://www.bioinformatics.org/wiki/Bio2RDF> [Accessed on 13th November 2012]

The Waterfall Model. Available at: <http://www.buzzle.com/articles/waterfall-model-advantages-and-disadvantages.html> [Accessed on 14th November 2012]

Guide to Harvard referencing. Available at: <http://libweb.anglia.ac.uk/referencing/harvard.htm> [Accessed on 17th November 2012]

Adam Connah (aoc9)

Tutorial on inserting RDF data into Virtuoso. Available at: <http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VirtRDFInsert> [Accessed on 7th November]