The EngD First Year Report

Ben Tagger

November 16, 2005

Preliminary EngD Title: A Framework for Biological Data Versioning

Starting Date: 27th September 2004

1st Supervisor: Anthony Finkelstein

2nd Supervisor: Chris Clack

Industrial Supervisor: Delmiro Fernandez-Reyes

Assessor: TBC

1 Introduction

This document provides a report of the major research to be undertaken during the course of the EngD programme. It also aims to include the requirements within the first year report form and, therefore, will include the form (as an appendix) with the relevant parts indicated on that form.

Section two aims to provide an introduction and overview of the chosen research problem. A preliminary literature review is contained in section three with a brief conclusion of the review at the end of the section. Section four will detail the contribution and scope of the proposed research during the EngD programme and section five will contain the proposed research activities needed to complete the proposed research. Section six will describe how the research is to validation and section seven provides a timetable of the planned research activities.

2 The Problem

2.1 Overview

The treatment of disease and illness is a rapidly developing technology. There are various methods for the diagnosis and treatment of cancer and other diseases and these methods have recently become more and more individualised. The early detection of disease has contributed significantly to the extent in which they can be successfully treated.

Traditional diagnosis techniques concentrate on identifying the patient's symptoms. However, the signs and symptoms of cancer (for example) may resemble those of another condition and are sometimes misleading. For example, a patient suffering from weight loss and abdominal pains may have stomach cancer or an ulcer. Traditional diagnosal techniques (for cancer) have included such procedures as biopsies. However, machine learning techniques have also recently been used for diagnosis. For example, breast cancer can be diagnosed from just one simple procedure when combined with the power of machine learning techniques, see [1].

Machine learning has also been used for some data-intensive applications. As an example of one such application, consider the use of gene-expression microarrays[1]. These microarrays make it possible to measure the rate at which cell (or portion of tissue) is expressing each of its thousands of genes. Microarrays can be used to;

- Infer regulatory pathways in cells.
- Identify novel targets for drug design.

[1]These are commonly referred to as "Gene Chips".

- Improve the diagnosis, prognosis and treatment planning for those suffering from disease.

This process generates a vast volume of data, which would take too long to manually analyse. This large volume of data must therefore be analysed automatically and this presents an opportunity for machine learning. Supervised, semi-supervised and unsupervised learning methods can be used to interpret large amounts of data relatively quickly. For this reason, biological data and machine learning techniques are being used together for a variety of medical applications (diagnosis, prognosis and drug response prediction).

2.2 The Problem with Biological Data

Data integration of geographically dispersed, heterogenous, complex biological databases is a key research area [13]. One of the principle issues of data integration is the data format. Ideally, a simple, self-describing format is best. However, many current biological databases provide data in flat files which are poor data exchange formats. Worse still, each biological database has a different format.

As is true of most scientific data, biological data possess the following characteristics.

- Complexity
- Incompleteness
- Error-prone

To add to these less than ideal characteristics, there exists a collossal amount of biological data which is stored in a variety of formats in a multitude of heterogenous systems. Scientists need an integrated view of these heterogenous data sources with advanced data accessing, analysing and visualisation tools [14]. Accessing the relevant data, combining data sources and coping with their distribution and heterogeneity is a tremendously difficult task. The continuing data growth will lead to an increasing need for large-scale data management. Biological discovery depends, to a large extent, on the presence of a clean, up-to-date and well-organised dataset [3].

However, one of the biggest problems to overcome is that of the semantics of the data. Although there exists considerable issues with the information currently held in the data, there are also problems arising from the experimental information that is not being retained as part of the data (or metadata). One of the distinctions of biological data with respect to other types of scientific data, is the complexity and variety of the experiments that yield the data. Moreover, this complexity and variety have influences over the data generated, but are not recorded completely in the metadata. Metadata is a description of the content, quality, lineage, contact, condition and other characteristics

of data[2]. The aim is to retain information about the data, which is important but which is not reflected by the data itself. In the case of biological data, the complexity and heterogeneity of the experiments (and, therefore, the required metadata) is very substantial and this results in an increased difficulty in data management.

2.3 What is Needed?

Given the vastness, its heterogeneity and other characteristics of biological data, the management of such data has become a key research area. To date, much of the research has focused on dealing with the vastness, heterogeneity and multiple sources of the data. Much of this work has been centered on improving the integration of the data, concentrating on data formats and standards. In contrast, little by comparison has been done on one aspect of data management, namely the versioning of biological data. But why is there a need for the versioning of life sciences data?

Consider the generation of life sciences data. Not only is the source data vast, heterogeneous and disparate, but the experiments that yield further results are complex and highly varied. Furthermore, with the advent of high-throughput computation, thousands of experiments can now be conducted in the same time it took to do only a handful several years ago (manually). Experiments may be repeated so as to verify results or to update data that has been subsequently updated itself. E-scientists may want access to previous versions of results in order to check repeatability or to verify other items of data. Previous versions of data may be used to check similarity with updated results or simply to re-run older experiments.

3 Reviewed Literature

This section contains a preliminary review of the literature relevant to the problem of biological data versioning. We discuss some varieties of biological data formats, followed by the issues of accessing, annotation and integration of the data. Subsection 3.3 describes some aspects and examples of scientific data versioning with a description of a framework that purports to track biological experiments. This section continues to review ontologies, the semantic web and data provenance.

3.1 Biological Formats

Sequence formats are the way in which biological data such as amino acid, protein and DNA sequences are recorded in a computer file. If you wish to submit some data to a data source, it must be in the format that that particular data source is prepared to

[2]According to the National Biological Information Infrastructure website - http://www.nbii.gov/datainfo/metadata/

receive. There are numerous sequence formats in the biological domain, too many to discuss their details and differences in this document. In general, each database will have its own sequence format. Some databases such as GenBank, EMBL and DDBJ (DNA Data Bank of Japan) share similar data formats (albeit with differing headers).

The EBI (European Bioinformatics Institute) ensures that all the information from molecular biology and genomic research is made publicly and freely available in order to promote scientific progress. EBI's databases and tools allow biological information to be stored, integrated, searched, retrieved and analysed. As mentioned above, each available database will require the appropriate tool in order to make successful queries and each database will require data to be submitted in the correct format.

Microarrays are one of the most important breakthroughs in experimental life sciences. They allow snapshots to be made of gene expression levels at a particular genomic stage[3]. Microarray data can be accessed though *Array Express* which is the public repository for microarray-based gene expression data. Although many significant results have been derived from microarray studies, one limitation had been the lack of standards for presenting and exchanging such data [7]. MIAME (Minimum Information About a Microarray Experiment) [7] describes the minimum amount of information necessary to ensure that the microarray data can be easily verified and enable the unambiguous interpretation and reproduction of such data.

The effective and efficient delivery of health care requires accurate and relevant clinical information. The storage of clinical information has traditionally been limited to paper, text or digitised voice. However, a computer cannot manipulate data in these formats. Clinical terminologies are also large, complex and diverse with respect the nature of medical information that has been collected over the 130 years of the discipline. There are numerous schemes which have been successful in supporting the collation and comparison of medical information. However, the problem arises when we try to transfer information between schemes. It is also hard to re-use schemes for purposes other than which they originally developed and this causes the proliferation of even more schemes. Galen (and the open source version OpenGalen [8]) provides a formal model of clinical terminology.

Mass spectrometry is a powerful analytical technique that is used to identify unknown compounds and quantify known compounds (even in very minute quantities). It is also used to establish the structure and chemical properties of molecules. Mass spectrometry can be used to sequence biopolymers (such as proteins and oligosaccharides), determine how drugs are used by the body and perform forensic analyses such as drug abuse, athlete steroid abuse among others. Due to the large amounts of information that can be generated by mass spectrometry, computers are essential (not only to control the mass spectrometer), but for spectrum acquisition, storage and presentation. Tools are available for spectral quantitation, interpretation and compound identification via on-line spectral libraries.

[3]Taken from http://www.ebi.ac.uk/Databases/microarray.html

3.2 Data Access, Annotation and Integration

Schönback et al [9]defines such a biological data warehouse as a subject-oriented, integrated, non-volatile, expert-interpreted collection of data in support of biological data analyses and knowledge discovery. BioWare, A framework for the construction of a biological data warehouse, is provided by [3]. As opposed to many SSDWs (Specialised Sequence Data Warehouse(s)), the users of BioWare do not require any specific programming or database expertise in order to retrieve, annotate and publish their data in the form of a searchable SSDW. However, BioWare does have some limitations. Given that the SSDW is designed to accessed via HTTP, a bottleneck is created with excessive downloading from public data sources. The current system, therefore, holds only 1000 entries. The data is still held in flat files, which impedes the speed of data access.

The EnsMart system [10] purports to provide a generic data warehousing system for fast and flexible access to biological datasets as well as integration with third-party data and tools. EnsMart is a system that is capable of organising data from individual databases into one query-optimised system. The generic nature of EnsMart allows the integration of data in a flexible, efficient, unified and domain-independent manner [10]. EnsMart is a self-contained addition to Ensembl software and data, providing access to commonly used parts of the genome data. One of the noted benefits of the EnsMart system is the ability to engineer your own version (with a little informatics expertise). This is useful is you wish to use data and queries that are not explicitly offered by the website. This will also overcome the problem of slowness or 'out-of-service' at www.ensembl.org when mining data in large quantities.

In the beginning, bioinformatics data was stored in flat files as ASCII characters. This was sufficient as the number of known proteins were small and there was little thought of genomic data. With the advent of rapid gene sequencing, it was established that techniques such as dynamic programming would be needed to cope with the exponential growth of data [11].

Ensembl [12] began as a project to automatically annotate vertebrate genomes. This basically involves running many commonly-used bioinformatics tools and combining the outputted data to produce genome-wide data sets such as protein-coding genes, genome-genome alignments and other useful information. Obviously, there is a considerable amount of processor power needed to produce the annotation for the vast amount of biological data. It is not feasible to compute this kind of problem on a single-CPU, but rather employ a distributed machine infrastructure.

Data integration of geographically dispersed, heterogenous, complex biological databases is a key research area [13]. One of the principle issues of data integration is the data format. Ideally, a simple, self-describing format is best. However, many current biological databases provide data in flat files which are poor data exchange formats. Worse still, each biological database has a different format. Bio2X [13] is a system that gets around this problem by converting the flat file data into highly hierarchical XML data

using rule-based machine learning techniques.

Data integration consists of wrapping data sources and either loading the retrieved data into a data warehouse or returning it to the user. *Wrapping* a data source means getting data from somewhere and translating it into a common integrated format [14]. There have been many systems designed for the specific purpose of biological data integration (too many to mention here). None of these approaches provides a seamless integration that permits the use of metadata about source content and access cost to permit the optimisation of the evaluation of the queries [14].

As the amount of biological data increases and becomes more pervasive, there arises various issues surrounding data quality, data legacy, data uniformity and data duplication. *Data cleaning* is the process by which biological data is corrected and standardised. However, due to the complex and diverse nature of the data, the problem of improving the data quality is non-trivial [15]. If quality of the data is not maintained, then the processes and operations that rely on this data will either fail or (arguably worse) provide skewed results. BIO-AJAX [15] is a toolkit that provides an extensible framework with various operations to address data quality issues through data cleaning within biological data repositories.

3.3 Scientific Data Versioning

Metadata is a description of the content, quality, lineage, contact, condition and other characteristics of data[4]. Biological metadata works in the same way. Information about the data, including content, quality, and condition makes the data more easily accessible to scientists and researchers. Unfortunately, conventional versioning systems do not efficiently record large numbers of versions. In particular, versioned metadata can consume as much space as the data itself [16]. Two space-efficient metadata structures for versioning file systems are examined in [16].

An interesting example of successful data versioning in a scientific domain is presented by Barkstrom [17]. According to [17], large-scale scientific data production for NASA's Earth observing satellite instruments involves the production of a very vast amount of data from a very wide variety of data sources[5].It is noted that while software versioning requires tracking changes principally in the source code, versioning of the data requires the tracking of changes in the source code, the data sources and the algorithm coefficients (parameters). Barkstrom [17] notes that changes in any of these items can induce scientifically important changes in the data.

Shui et al. [18] present an XML-based version management system for tracking complex biological experiments. Shui et al. [18] make a good case for the need for biological

[4]According to the National Biological Information Infrastructure website - http://www.nbii.gov/datainfo/metadata/

[5]In fact, Barkstrom [17] quotes the production values at tens of thousands of files per day from tens or hundreds of different data sources.

data versioning, similar to the one made in this report. The framework uses generic versioning operations (insert, delete) and defines three more (update, move and copy) in order to describe changes in the XML. The framework can store every single component of an entire experiment in XML. A change to any component will result in a new version. Users can then query the system to return sets of data so they can see the differences between sets of results according to the materials used. The title of the publication [18] reports to track complex biological experiments. However, the conclusion of the same publication states clearly that the framework only tracks changes to laboratory based data. This publication is important in assessing the need for data versioning of complex life-science data and experimentation, but proposes a framework that appears to deliver little more than generic XML versioning, applied in a scientific data context.

3.4 Ontologies and the The Semantic Web

There is a large amount of heterogeneous biological data currently available to scientists. Unfortunately, due to its heterogeneity and the widespread proliferation of biological databases, the analysis and integration of this data represents a significant problem. Biological databases are inherently distributed because the specialised biological expertise that is required for data capture is spread around the globe at the sites where the data originate. To make the best use of this data, different kinds of information must be integrated in a way that makes sense to biologists.

Biologists currently waste a great deal of time searching for available information in various areas of research. This is further exacerbated by the wide variations of terminology used by different researchers at different times. An ontology provides a common vocabulary to support the sharing and reuse of knowledge.

The Gene Ontology (GO) project provides structured, controlled vocabularies and classifications that cover several domains of molecular and cellular biology. The project is driven and maintained by the Gene Ontology Consortium. Members of this consortium continually work collectively and, with the help of domain experts, seek to maintain, expand and update the GO vocabularies. Collaborations of this nature are difficult to maintain due to geography, misunderstandings and the length of time it takes to get anything done.

Ontologies are considered the basis building blocks of the Semantic Web as they allow machine supported data interpretation reducing human involvement in data and process integration [19]. Ontologies provide a reusable piece of knowledge about a specific domain. However, these pieces of knowledge are often not static, but evolve over time [20]. The evolution of ontologies causes operability problems, which hamper the effective reuse. Given that these changes occur and given that they are occurring within a constantly changing, decentralised and uncontrolled environment such as the web, support to handle these changes is needed. This is especially prudent with respect to the semantic web, where computers will be using the data. The semantic web requires inter-

operability on the semantic level. *Semantic interoperability* requires standards not only for the syntactic form of documents, but also for the semantic information.

XML (Extensible Markup Language) and RDF (Resource Description Framework) were the current standards for establishing semantic interoperability on the Web. However, XML only describes document structure. RDF better facilitates interoperation because it provides a data model that can be extended to address sophisticated ontology representation techniques [21]. XML is intended as a markup-language for arbitrary document structure. An XML document consists of a properly nested set of open and close tags, where each tag can have a number of attribute-value pairs. One of the important aspects of XML is that the vocabulary is not set, but rather can be defined per application of XML.

In particular, XML falls down on the issue of scalability. Firstly, the order in which elements appear in an XML document is significant and can change the meaning of the document. When it comes to semantic interoperability, XML has disadvantages. Since XML only deals with the structure of the document, there is no way of recognising or extracting semantic meaning from a particular domain of interest.

3.5 Data Provenance

The widespread nature of the Internet and the ease with which files and data can be copied and transformed has made it increasingly difficult to determine the origins of a piece of data. The term *data provenance* refers to the process of tracing and recording the origins of data and its movement between databases. With many kinds of data, provenance is not important. However, for scientists focussed on the *accuracy and timeliness* of the data, provenance is a big issue [22].

Provenance allows us to take a quantity of data and examine its *lineage*. Lineage shows each of the steps involved in sourcing, moving and processing the data [23]. In order to provide provenance, all datasets and their transformations must be recorded. Scientists are often interested in provenance because it allows them to view data in a derived view and make observations about its quality and reliability [22]. Goble [24] presents some notable uses for provenance:

1. *Reliability and quality:* Given a derived dataset, we are able to cite its lineage and therefore measure its credibility. This is particularly important for data produced in scientific information systems.

2. *Justification and audit:* Provenance can be used to give a historical account of when and how data has been produced. In some situations, it will also show why certain derivations have been made.

3. *Re-usability, reproducibility and repeatability:* A provenance record not only shows how data has been produced, it provides all the necessary information to reproduce the results. In some cases the distinction between repeatability and reproducibility

must be made. In scientific experiments results may be different due to observational error or processing may rely on external and volatile resources.

4. *Change and evolution:* Audit trails support the implementation of change management.

5. *Ownership, security, credit and copyright:* Provenance provides a trusted source from which we can procure who the information belongs to and precisely when and how it was created.

Zhao et al. [25] describes three further purposes for provenance[6] from the viewpoint of the scientist:

1. *Debugging:* Experiments may not produce the desired results. The scientist requires a log of events recording what services were accessed and with which data.

2. *Validity Checking:* If the scientist is presented with a novel result, he/she may wish to perform expensive laboratory-based experiments based on these results. Although positive that the workflow design is valid, the scientist may still want to check how this data has been derived to ensure it is worthy of further investigation.

3. *Updating:* If a service or dataset used in the production of a result has changed, the scientist will need to know what implications that change has on those results.

The field of molecular biology supports many hundreds of public databases, but only a handful of these can be considered to contain "source" data in that they receive experimental data. Many of the databases actually reference themselves. This sounds contradictory until you take into account that much of the value associated to a data source comes from the curation and annotation (conducted by experts) of the data.

Most implementors and curators of scientific databases would like to record provenance, but current database technology does not provide much help in this process as databases are typically rigid structures and do not allow the kinds of *ad hoc* annotations that are often needed to record provenance [22].

It is not only the biological science domain that is concerned with data provenance. Large-scale, dynamic and open environments such as the Grid and web services build upon existing computing infrastructures to supply dependable and consistent large-scale computational systems [26]. With both scientific experiments and business transactions, the notion of lineage and dataset derivation is of paramount importance since without it, information is potentially worthless.

Provenance needs to be stored and [26] envisages two possible solutions for storage.

1. Data provenance is held alongside the data as metadata.

2. Data provenance can be stored in a dedicated repository, made accessible as a Grid or web service.

[6]These are not entirely exclusive to those described by Goble [24].

The first solution requires the holders of any such data to maintain the integrity of the provenance records as transformations take place, and it imposes significant changes to any existing data storage structures. Such a kind of provenance can be useful for a user to remember how a result was derived, and what steps were involved. It is unclear that such provenance can be trusted by any third party, since the data and provenance owner has to be trusted to have recorded provenance properly.

Workflow enactment is the automation of a process during which documents, information or tasks are passed from one participant to another for action, according to a set of declarative or procedural rules [26]. In Grid applications, this task is often performed by a workflow enactment engine, which uses a workflow script, such as WSFL or BPEL4WS, to determine which services to call, the order to execute them in and how to pass datasets between them.

3.6 Conclusions from the Literature

Biological data has three principal (unfavourable) characteristics; high volume, disparate heterogeneous sources, and the complexity of the data semantics. Although the first two have received considerable attention [3, 10, 12], comparatively little effort has been made in managing the semantic complexity. There have been various attempts at investigating the versioning of scientific data [17, 18] but these have, so far, failed to address the aforementioned issues surrounding biological data specifically. It seems clear that a framework for versioning biological data will need to take into account the difficulties of that very data in order to prove successful.

4 EngD Contribution and Scope

The separation of the data and algorithms, as illustrated in Figure 1, presents an opportunity for applying a data management strategy. Moreover, given the aforementioned characteristics of biological data, the need for such data management is obvious. Versioning of the data is one aspect of data management and will be the focus of the project.

However, simply versioning the data itself is not sufficient. There is little point in retaining the data if there is no information about how it was generated (or obtained). Information about the experiment that yielded the data must also be retained and, therefore, versioned. This *metadata* must be versioned alongside the data, but exactly what information is needed?

An experiment will usually begin with a hypothesis or at least, less formally, an idea of the experimenter. It may be important to store this for anecdotal reasons or it may serve to validate the results. It may be useful as a natural language description of the experiment. The owner, hypothesis/idea and time of the experiment are recorded. The

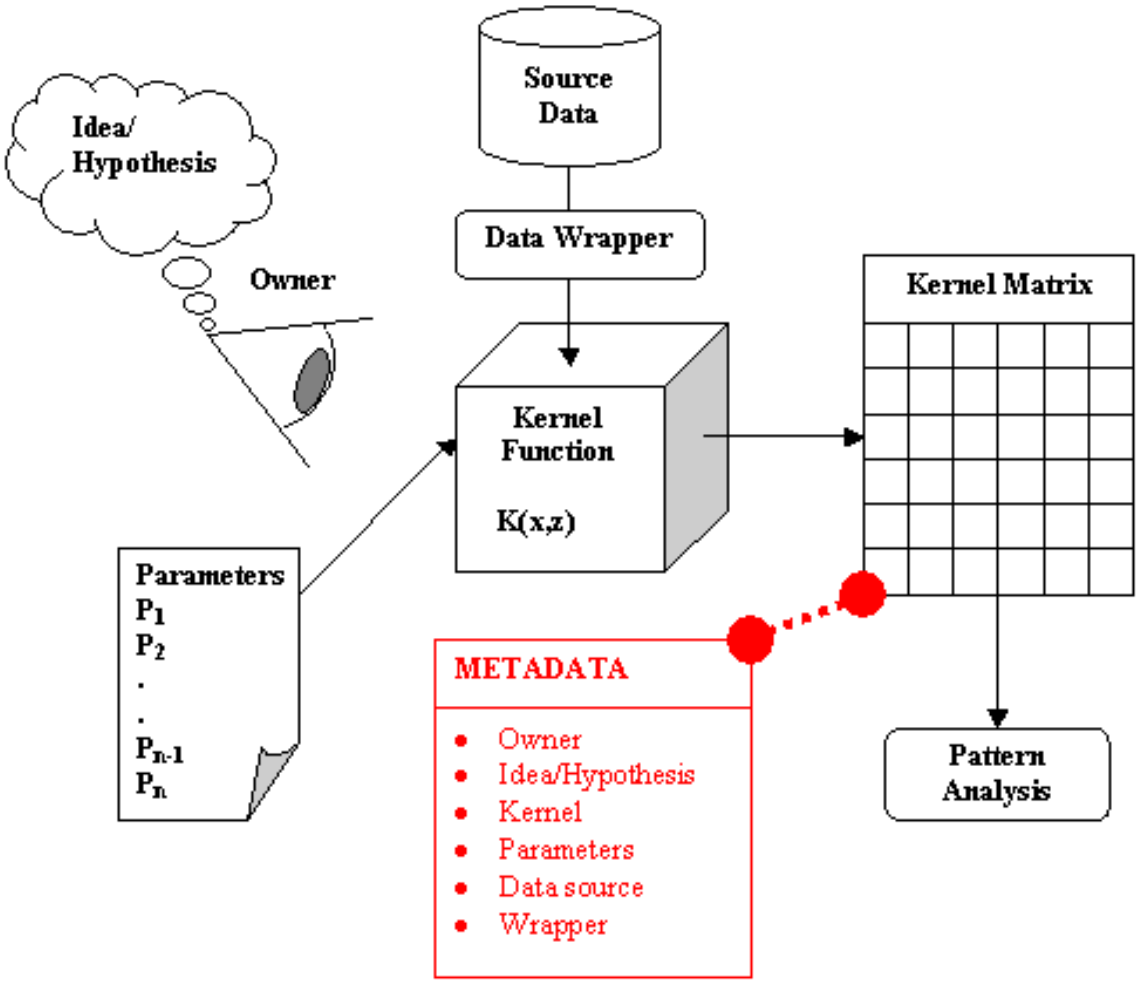

Figure 1: Some of the metadata involved in the application of kernel methods

data source may need to be *wrapped* before its use in the kernel function. Wrapping a datasource involves getting data from somewhere and translating it into a common data format [14]. This area of data integration has been widely researched and there are numerous wrapping tools currently available. Whether the wrapper is a third-party tool or is custom built by the e-scientist, it should be recorded. The kernel function and parameters are also recorded.

Note that the metadata (in red) describes the kernel matrix (and its inception) and bears no relation to later parts of the process such as the pattern analysis. It is this metadata (and, potentially, the kernel matrix) that is to be versioned. The data and metadata generated by the pattern analysis is also relevant to the experimentation process and, as such, may be versioned along side the kernel matrix versioning data.

Hopefully, the need for increased efforts in data management and versioning for the life sciences is now clear. Although a considerable amount of research has been focused on various aspects of life sciences data management, little has been done specifically on versioning. The goal of the project is to develop a framework that can be used for the versioning of life sciences data. The versioning framework will initially be applied in a machine learning context. Upon successful completion of this part and if sufficient time is available, the possibility for a more generic framework for the versioning of life sciences data will then be considered.

5 Proposed Research Activities

This section will describe the various research activities that will take place in order to successfully achieve the contribution described in the above section. The research activities have been grouped into one of several groups; Requirements Analysis, Background Research, Implementation, Validation and Documentation. It should be noted that this section aims to give only a preliminary outline of activities, rather than an extensive and detailed description of progress.

1. Requirements Analysis: What data is to be versioned?

If we are to address the versioning of scientific data, it is important to first ascertain exactly what (kinds of) information is to be versioned. Only once the nature of the data is discovered, can the question of how to version the data be attempted. The source of this information will be primarily the scientists involved in the experimentation process. This activity is scheduled to occur at the very beginning of the activities. This is due to there being other activities that rely on the outcomes of this activity. The discovery of the exact information that is to be retained and versioned is beyond the scope of this document. Given the complexity and variety of the data (as mentioned before), it is likely that not all available data will be versioned. Refer to Figure 2 for a look at the kinds of information that may be versioned.

There will certainly be additional items of information to be versioned and much more detail in the above items than is described here. The source for this information will be, primarily, the scientists involved in the experimentation process. Once the versioning information has been identified, it is necessary to ask the question of how versioning is to be applied. It is also important to consider the point(s) at which the versioning is to be applied.

2. Requirements Analysis: Presentation Requirements

The purpose of this activity is to elicit all the remaining requirements from the experimenter. Broadly, it is aimed at discovering how the experimenter wants the framework to version his/her data. How do they want the versions to be made available to them? What operations are to be performed on the versions and how does the experimenter

want to interact with them? This activity represents a considerably complex task and is likely to consist of several smaller activities. The outcome of this activity in its entirety will be a requirements specification document, against which the implementation may be validated.

This period of requirements analysis is likely to continue well into the project. Once implementations become available, the requirements invariably change. While a limited degree of change is desirable and even necessary, most of the requirements gathering is to be conducted before the main implementation stages.

3. Background Research: Investigate Experimentation Process

In order to start the implementation of a versioning framework for some experimental process, it is necessary to first become familiar with that process. In order to understand how to handle the experimental data, it is necessary to understand the format of the data and also the nature of the experimentation process in order to understand at which point in the process the data is currently held.

4. Background Research: Object Serialisation, Castor and Others

A considerable part of the project will be focussed on the storage of data and how to store and retrieve versions of data. Castor JDO provides methods for translating Java objects to XML (and vice-versa). The Java Data Objects (JDO) API is a standard interface-based Java model for directly storing Java domain model instances into a persistent store (database). Some background research needs to be carried out to investigate the various methodologies available for this application.

5. Implementation: Accessing the Relevant Data

One of the first stages in the implementation process will be handling of the relevant data. By the time this activity starts, the information that is required for versioning will be already determined (see activity 1). The investigation of the experimentation process (activity 3) will also be in progress and will be necessary in order to understand how to get a handle on the relevant experimentation data. See figure 3 for a very simplified view of the implementation process. The expected outcome of this research activity is some sort of automated, structured metadata recording process, whereby versions of data and metadata are automatically generated throughout the experimentation process.

6. Implementation: Storage and Retrieval of Versions

Once there is a handle on the relevant data, the implementation process will focus on the storage and retrieval of these versions. It is intended that this activity occur in tandem with activity 4, which broadly provides the background for this stage of implementation.

7. Implementation: Version Presentation

This activity will largely depend on the outcome of the requirement analysis documented in activity 2. Activity 2 should uncover the levels of functionality that the system

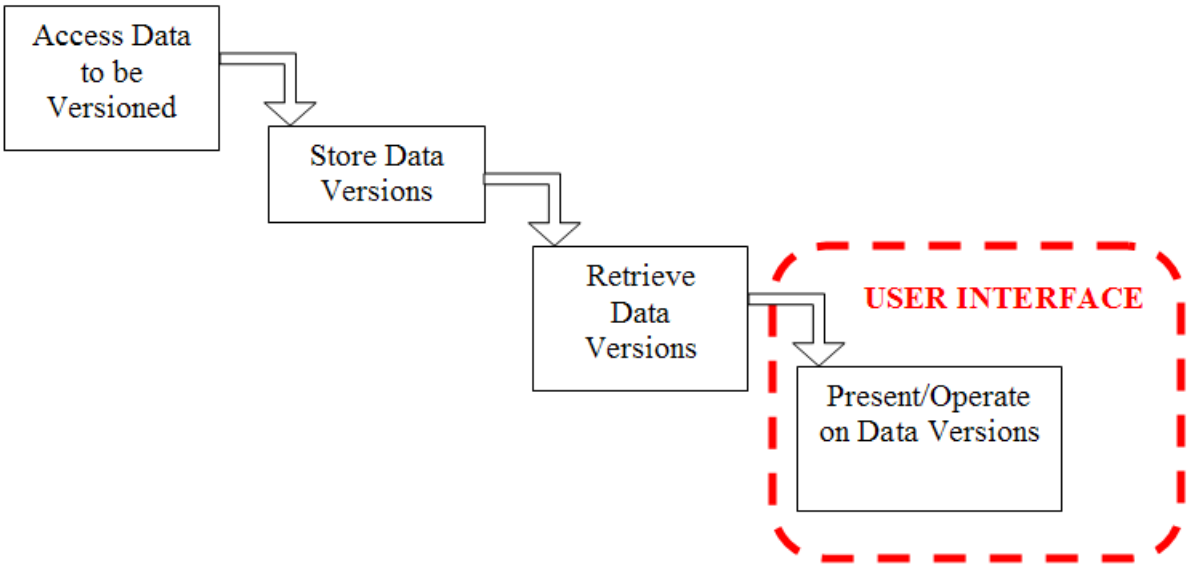

Figure 2: A simplified view of the implementation process

will be required to achieve and the implementation of this functionality as well as the presentation of that functionality will constitute this activity.

8. Implementation: Framework Integration

This activity represents the efforts of integrating the framework back into the experimentation process. Ideally, the framework should operated transparently from the experimentation process so that there will be only minimum impact on the usability of the existing process. There are two periods of framework integration realised on the timetable (see figure 4). These represent the two occurrences where the implementation interacts with the existing experimentation process. Both activities 5 and 7 require integration with the experiment process.

9. Documentation: Ongoing Documentation

It is hoped that documentation shall be ongoing over the course of the project. This will help improve the quality of the documentation[7] and will be easier to compile when needed. This documentation may also help provide the validation for the project.

10. Documentation: Requirements Specification

A requirements specification for a biological data versioning framework will be produced by the end of year two.

[7]Documentation written nearer the time of the subject matter will normally be of higher quality as recollection generally does not improve with time.

6 Validation

Broadly speaking, the contribution will be validated against the progress of the above research activities. A requirements specification document will be produced in the latter stages of year 2 (see activity 10 and timetable). This document will aim to capture all the relevant requirements for the versioning framework. Validation for the contribution can be measured against the successful implementation of the requirements set out in the specification document.

7 Plan

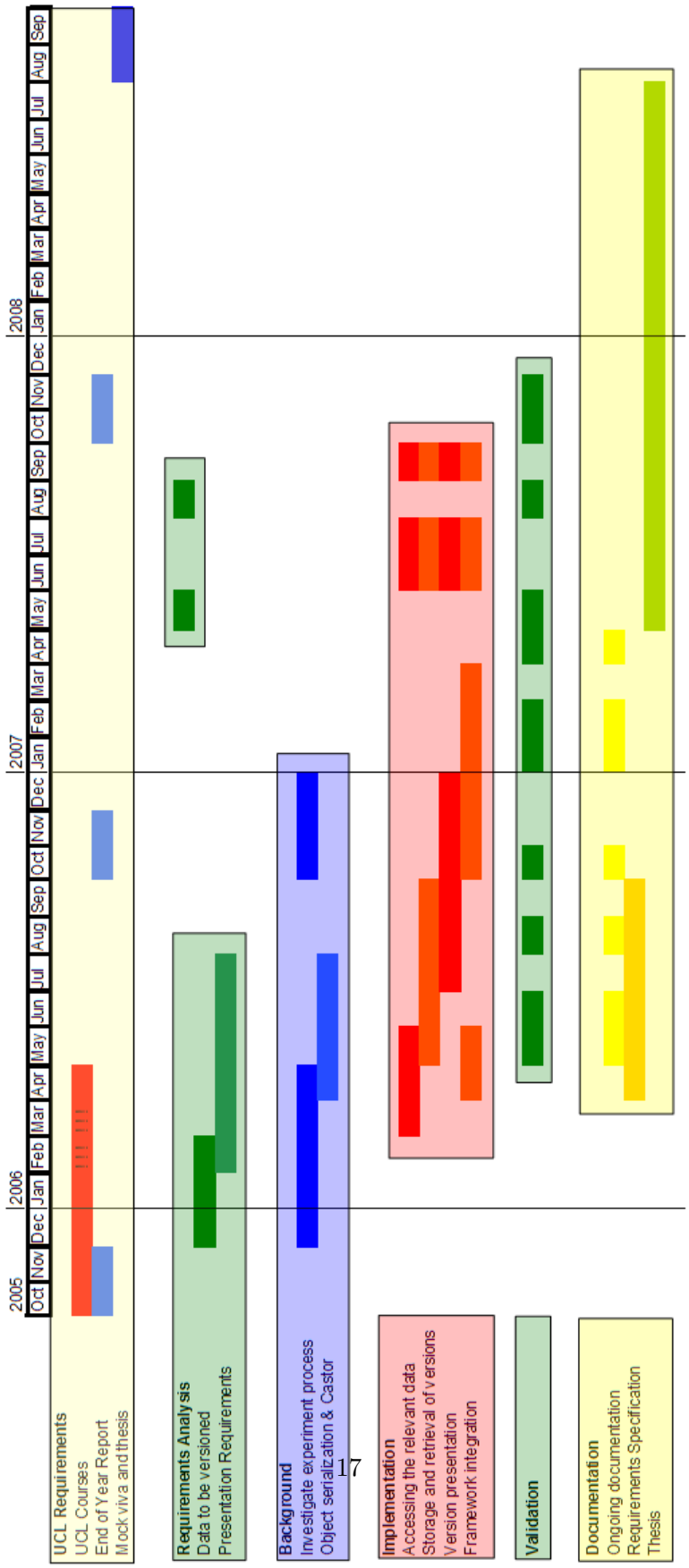

Figure 3: Project plan for the next three years

References

[1] W.H. Wolberg, W.N. Street, O.L. Mangasarian. Machine learning techniques to diagnose breast cancer from image-processed nuclear features of fine needle aspirates. *Cancer Letters 77:163-171*, 1994.

[2] Z. Lacroix, Biological data integration: Wrapping data and tools. *Ieee Transactions on Information Technology in Biomedicine, vol. 6, pp. 123- 128,* 2002.

[3] J. Koh et al. BioWare: A framework for bioinformatics data retrieval, annotation and publishing. *SIGIR 2004 Workshop*

[4] S. Yang, S. S. Bhowmick, and S. Madria. Bio2X: a rule-based approach for semi-automatic transformation of semi-structured biological data to XML. *Data and Knowledge Engineering, vol. 52, pp. 249-271,* 2005.

[5] T. De Bie, N. Cristianini. Kernel methods for exploratory data analysis: a demonstration on text data. *Proc. of the joint IAPR international workshops on Syntactical and Structural Pattern Recognition (SSPR 2004) and Statistical Pattern Recognition (SPR 2004), Lisbon*, August 2004.

[6] J. Shawe-Taylor, N. Cristianini. Kernel methods for pattern analysis *Cambridge University Press, Cambridge*, 2004.

[7] A. Brazma et al. Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Nat Genet, vol. 29, pp. 365-71*, 2001.

[8] A. L. Rector, J. E. Rogers, P. E. Zanstra, and E. Van Der Haring. OpenGALEN: open source medical terminology and tools. *AMIA Annu Symp Proc, pp. 982*, 2003.

[9] C. Schönbach, P. Kowalski-Saunders, and V. Brusic. Data warehousing in molecular biology. *Brief Bioinform, vol. 1, pp. 190-8*, 2000.

[10] A. Kasprzyk, D. Keefe, D. Smedley, D. London, W. Spooner, C. Melsopp, M. Hammond, P. Rocca-Serra, T. Cox, and E. Birney. EnsMart: A generic system for fast and flexible access to biological data. *Genome Research, vol. 14, pp. 160-169*, 2004.

[11] J. A. Cuff, G. M. P. Coates, T. J. R. Cutts, and M. Rae. The Ensembl computing architecture. *Genome Research, vol. 14, pp. 971-975*, 2004.

[12] T. Hubbard. Ensembl 2005. *Nucleic Acids Research, vol. 33, pp. D447-D453* 2005.

[13] S. Yang, S. S. Bhowmick, and S. Madria. Bio2X: a rule-based approach for semi-automatic transformation of semi-structured biological data to XML. *Data and Knowledge Engineering, vol. 52, pp. 249-271*, 2005.

[14] Z. Lacroix, Biological data integration: Wrapping data and tools. *Ieee Transactions on Information Technology in Biomedicine, vol. 6, pp. 123-128*, 2002.

[15] K. G. Herbert, N. H. Gehani, W. H. Piel, J. T. L. Wang, and C. H. Wu. BIO-AJAX: An extensible framework for biological data cleaning. *Sigmod Record, vol. 33, pp. 51-57*, 2004.

[16] Craig A. N. Soules, Garth R. Goodson, John D. Strunk, and Greg Ganger. Metadata efficiency in versioning file systems. *Conference on File and Storage Technologies (San Francisco, CA).* 31 March–02 April 2003.

[17] B. R. Barkstrom. Data product configuration management and versioning in large-scale production of satellite scientific data. *Software Configuration Management, vol. 2649, pp. 118-133*, 2003.

[18] W. M. Shui, N. Lam, R. K. Wong. A Novel Laboratory ersion Management System for Tracking Complex Biological Experiments. *bibe, p. 133, Third IEEE Symposium on BioInformatics and BioEngineering (BIBE'03)*, 2003.

[19] J. Cardoso and A. Sheth. Introduction to semantic web services and web process composition. *Semantic Web Services and Web Process Composition, vol. 3387, pp. 1-13*, 2005.

[20] M. Klein and D. Fensel. Ontology versioning for the Semantic Web. *In Proceedings of the International Semantic Web Working Symposium (SWWS), Stanford University, California, USA*, July 30 – Aug. 1, 2001.

[21] S. Decker, S. Melnik, F. Van Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann, and I. Horrocks. The semantic Web: The roles of XML and RDF. *Ieee Internet Computing, vol. 4, pp. 63-74*, 2000.

[22] P. Buneman, S. Khanna, and W. C. Tan. Data provenance: Some basic issues. *Fst Tcs 2000: Foundations of Software Technology and Theoretical Computer Science, Proceedings, vol. 1974, pp. 87-93*, 2000.

[23] D. Pearson. Data requirements for the grid. *Scoping Study Report.* February 2002. Status Draft.

[24] C. Goble. Position statement: Musings on provenance, workflow and (semantic web) annotations for bioinformatics. *Data provenance/derivation workshop.* October 2002.

[25] J. Zhao, C. Wroe, C. Goble, R. Stevens, D. Quan, and M. Greenwood. Using semantic web technologies for representing e-Science provenance. *Semantic Web - Iswc 2004, Proceedings, vol. 3298, pp. 92-106*, 2004.

[26] M. Szomszor and L. Moreau. Recording and reasoning over data provenance in web and grid services. *On the Move to Meaningful Internet Systems 2003: Coopis, Doa, and Odbase, vol. 2888, pp. 603-620,* 2003.