Metadata-Driven Management of Scientific Data

Thesis

Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in
the Graduate School of The Ohio State University

By
Aman Kumar

Graduate Program in Computer Science and Engineering
The Ohio State University
2009

Thesis Committee:
Prof. Rajiv Ramnath, Advisor
Prof. Michael Freitas
Prof. Jayashree Ramanathan
Abstract

Management of large amounts of complex and multi-dimensional scientific data, residing in different data sources has become a huge challenge in the scientific domain. Researchers require technologies to facilitate understanding of the data, in particular the ability to effectively organize complex data, perform efficient analysis, share and discover relevant data and facilitate searches over large data sets. Existing systems and approaches have tried to resolve these problems individually and have been successful to some extent. But there is a lack of an end-to-end solution that suits the dynamically emerging needs of the scientific community. This thesis is an attempt to fill this gap by presenting the work done in each of these areas and by proposing an end-to-end framework that follows the meta-data driven approach to resolve these issues in an integrated manner. Also, the framework demonstrates that the concept of ontologies and folksonomies can be leveraged to manage scientific data. To provide validation to the framework, the component architecture and functionality of its simple yet powerful implementation, MetaDB is described.
Dedication

Dedicated to my father
Acknowledgements

I would like to express my deep and sincere gratitude to my advisor Professor Rajiv Ramnath. His immense knowledge and critical thinking process have been of great value for me. I thank him for his patience and encouragement.

I also wish to express my warm and sincere thanks to Professor Michael Freitas for his continued support, insights and suggestions that helped to shape my research. I could not have imagined having a better mentor for my Master’s thesis.

Last but not the least, I would like to thank Professor Jayashree Ramanathan for her guidance and support and helping me out whenever I needed it.
Vita

2003………………………………Bachelor of Technology, TIT&S, Bhiwani, India


August 2004 – March 2006 ……….. Senior Software Engineer, Quark India Pvt Ltd.

March 2006 – August 2007 …………… Senior Software Engineer, GlobalLogic Pvt Ltd.

Fields of Study

Major Field: Computer Science and Engineering
# Table of Contents

Abstract ........................................................................................................................................ ii

Dedication .................................................................................................................................. iii

Acknowledgements ....................................................................................................................... iv

Vita ....................................................................................................................................... v

Table of Contents ....................................................................................................................... vi

List of Tables ............................................................................................................................... xii

List of Figures ............................................................................................................................. xiii

Chapter 1: Problem Statement ................................................................................................. 1

1.1 Data Management ............................................................................................................... 2

1.2 Data Management in Scientific Domain ........................................................................... 5

1.3 What is Needed: the Big Picture ....................................................................................... 8

1.4 Issue in Data Management ............................................................................................... 12

1.4.1 Uniform Access to Data and Interoperability Support ............................................... 13

1.4.2 Providing Meaning to Data ......................................................................................... 13

1.4.3 Data Consolidation and Integration ............................................................................ 14
1.4.4 Data Ownership and Security ................................................................. 14
1.4.5 Knowledge and Information Discovery .................................................. 14
1.4.6 Pipelining and Provenance .................................................................... 15
1.4.7 Extensibility and Flexibility .................................................................. 15
1.4.8 Providing an Intuitive User Interface (Folksonomies) ............................ 16

Chapter 2: Related Research ............................................................................. 17

2.1 Data Management Systems ........................................................................ 17
2.1.1 Scientific Data Libraries ......................................................................... 20
2.1.1.1 National Center for Biotechnology Information ................................. 21
2.1.1.2 NetCDF (Network Common Data Form) ............................................. 22
2.1.1.3 Model Data System (MDSplus) .......................................................... 23
2.1.2 Standalone Metadata Management Systems ........................................... 23
2.1.2.1 Metadata Catalog Service (MCS) ......................................................... 24
2.1.2.2 Data Warehouse Systems .................................................................. 25
2.1.2.3 Laboratory Information Management System (LIMS) ..................... 26
2.1.3 Data Grids ............................................................................................. 26
2.1.3.1 caGrid ............................................................................................... 27
2.1.3.2 Storage Resource Broker (SRB) ......................................................... 27
2.2 Discussion .......................................................................................................... 28

Chapter 3: The Metadata Approach.............................................................................. 32

3.1 What is Metadata................................................................................................ 32

3.2 Ontology............................................................................................................. 34

3.3 Folksonomy........................................................................................................ 36

3.4 Why Metadata .................................................................................................... 36

3.4.1 Metadata is Pervasive ................................................................................. 37

3.4.2 Enables Data Independence ........................................................................ 40

3.4.2.1 Physical Data Independence ................................................................ 41

3.4.2.2 Logical Data Independence ................................................................. 41

3.4.3 Unifying Approach ..................................................................................... 42

3.5 Categorizing Metadata ....................................................................................... 42

3.6 Applying Metadata............................................................................................. 48

Chapter 4: Implementation: MetaDB ........................................................................... 51

4.1 Component Architecture .................................................................................... 51

4.2 Enabling Technologies....................................................................................... 57

4.3 Managing Data Using MetaDB – A Real Example ........................................... 60

Chapter 5: Conclusion and Future Work ...................................................................... 71
References............................................................................................................................................ 73

Appendix A: MetaDB – Initial User Stories......................................................................................... 77

A.1 Generating RAW Data.................................................................................................................... 77
A.1.1 Exiting........................................................................................................................................ 77
A.1.2 Desired ...................................................................................................................................... 77
A.1.3 Notes ....................................................................................................................................... 78

A.2 Blanks and Standards Sample Data.............................................................................................. 78
A.2.1 Exiting........................................................................................................................................ 79
A.2.2 Desired ...................................................................................................................................... 79

A.3 Login and Data Organization............................................................................................................ 79
A.3.1 Exiting........................................................................................................................................ 80
A.3.2 Desired ...................................................................................................................................... 80
A.3.3 Notes ....................................................................................................................................... 82

A.4 Adding Notes to Data....................................................................................................................... 82
A.4.1 Exiting........................................................................................................................................ 82
A.4.2 Desired ...................................................................................................................................... 82

A.5 Finding Experiment/Analysis Data.................................................................................................. 83
A.5.1 Exiting........................................................................................................................................ 83
A.11.1 Exiting ..................................................................................................... 88
A.11.2 Desired .................................................................................................... 88
A.12 Results conversion to Excel Format ............................................................... 88
  A.12.1 Exiting ..................................................................................................... 88
  A.12.2 Desired .................................................................................................... 89
A.13 Conversion of RAW Data Graphics using Template ................................. 89
  A.13.1 Exiting ..................................................................................................... 89
  A.13.2 Desired .................................................................................................... 89
List of Tables

Table 1: data Lifecycle - Metadata ................................................................. 40

Table 2: Example Metadata – Categorized ................................................... 47
List of Figures

Figure 1: Scientific Data Lifecycle – Existing................................................................. 9
Figure 2: Scientific Data Lifecycle - Desired ................................................................. 11
Figure 3: issues in scientific data management.............................................................. 12
Figure 4: Data/Metadata confusion............................................................................. 33
Figure 5: Data Lifecycle ............................................................................................... 37
Figure 6: Categorizing Metadata ............................................................................... 43
Figure 7: Metadata Levels ........................................................................................... 46
Figure 8: Logical Collections...................................................................................... 49
Figure 9: MetaDB - Complete Architecture ................................................................. Error! Bookmark not defined.
Figure 10: MetaDB - Implemented Components.......................................................... 54
Figure 11: Dynamic Architecture Flow ....................................................................... 56
Figure 12: MetaDB - Database Schema......................................................................... 59
Figure 13: Example - Project Size .............................................................................. 61
Figure 14: Example - Project Structure ....................................................................... 62
Figure 15: Example - Project Heirarchy ...................................................................... 63
Figure 16: Example - Create Tag................................................................................ 64
Figure 17: Example - Add Tag .................................................................................... 65
Figure 18: Example - Multiple Views ................................................................. 66
Figure 19: Example - Multiple Views 2 ................................................................. 67
Figure 20: Example - Multiple Tag Selection....................................................... 68
Figure 21: Example - Different Aspects 1 ............................................................ 69
Figure 22: Example - Different Aspects 2 ............................................................ 70
Chapter 1: Problem Statement

Noted futurist and best-selling author of Megatrends [1] Naisbitt said almost 25 years back, "we are drowning in information, but we are starved for knowledge. Uncontrolled and unorganized information is no longer a resource in an information society, instead it becomes the enemy." This statement is as valid today as it was at that time. Data management systems are still evolving and are trying to catch up with the ever increasing size and complexity of data. This also reminds us of a poem, Choruses by T.S. Eliot in his famous book ‘The Rock’ [2]:

"The endless cycle of idea and action,

Endless invention, endless experiment,

Brings knowledge of motion, but not of stillness,

Where is the Life we have lost in living?

Where is the wisdom we have lost in knowledge?
In their book Megatrends (1982), Naisbitt and Aburdene further observed that we were shifting from an Industrial Society to an Information Society [1]. This means that instead of industrial processes, information and data would take the center stage. The resource based view (RBV) of organizations and competencies perspectives highlight the reflection of this changing trend in the strategy arena [3]. Earlier, it was processes that were more important and information was created either as a side-effect or result of process execution to achieve a certain goal. In an Information Society, however, data, need of data and meaning of data derive the processes that would act on data to produce what is required.

1.1 Data Management

Both business and the scientific community believe organizations can gain and sustain a long-term competitive advantage by managing its data and thus creating knowledge. That is the reason that in recent years, data management has become a critical subject of discussion in both the communities. Although business as well as scientific communities are convinced of the potential that can be realized from knowledge resources, there is not a consensus about the characteristics of data and knowledge and the ways data management should be used. Researchers and academics have taken different
perspectives on data management, ranging from technological solutions to the communities of practices, and the use of the best practices [4].

Data management is very hard to define precisely and simply. People also use terms such as information management and knowledge management to refer to the same concept. So before defining data management, we need to first define data, information and knowledge itself. According to Russell Ackoff, a systems theorist and professor of organizational change, the content of the human mind can be classified into five categories [5, 6]:

- **Data:** symbols

- **Information:** data that are processed to be useful; provides answers to "who", "what", "where", and "when" questions

- **Knowledge:** application of data and information; answers "how" questions

- **Understanding:** appreciation of "why"

- **Wisdom:** evaluated understanding.
In general, data are considered as raw facts, information is regarded as an organized set of data, and knowledge is perceived as meaningful information. According to Albert Einstein, “Knowledge is experience. Everything else is just information. [7]” As different people refer to these terms loosely we will use these terms interchangeably in this paper.

Now as we are somewhat clear about what is data, let us try to find an appropriate definition for data management. According to Ganesh D. Bhatt data management is the process of knowledge creation, validation, presentation, distribution, and application. These five phases in knowledge management allow an organization to learn, reflect, and unlearn and relearn, usually considered essential for building, maintaining, and replenishing of core-competencies [4]. The official definition provided by Data Management Body of Knowledge (DAMA-DMBOK) is: "Data management is the development, execution and supervision of plans, policies, programs and practices that control, protect, deliver and enhance the value of data and information assets."

Although these definitions seem to capture the essence and motivation of data management, but they fail to provide any guidance as to what should be done to achieve effective data management. Problem of effective data management and sharing is ubiquitous. With the advent of large scale use of computers, we are collecting and storing large amount of data each day in almost every industry domain. Not only we are storing more data but the complexity of data has also increased. The ever increasing affordability of storage space has only compounded the problems. Also, with the rise of internet and
communication technologies, the need is being felt for more and more collaboration between different data sources. To share and consolidate information we need to have a standard way of accessing the data and common data formats. Web services resolved the first issue by providing a standard way to represent data access layers. However, Web services are just focused on physical connectivity, and do not address the issue of heterogeneous formats of the data being exchanged. Unfortunately, there is severe lack of standard data formats and most of the information that needs to be consolidated is heterogeneous. This makes data hard to share and vulnerable to miss-interpretation.

1.2 Data Management in Scientific Domain

In scientific domain, the problem is further compounded because of the nature of data and specific needs of the scientific community. Firstly, the data generated is usually more as compared to the business domain. Researchers are struggling to keep up with the ever increasing volumes of information, and information sources at our disposal. As the precision of scientific instruments have increased, scientific communities are generating more data per experiment [8]. This has caused an explosion of data that must be rapidly analyzed and consolidated with related data. Storing peta-scale data is no longer an issue as storage space has become more affordable. However, the processing power and I/O bandwidth hasn’t kept pace with the progress in storage capacity. Also, the algorithms to analyze and process scientific data have become complex [8]. They have evolved over time and have become more memory and time intensive. The problem is compounded
when algorithms with exponential running time and space requirements work on large
datasets. Biotechnology, proteomics, pharmaceuticals, etc. researchers are constantly
struggling to keep up with volumes of information available.

Second, data that the researchers and scientist have to deal with in scientific domain is
much more complex. Scientific experiments and analysis often generate multi-
dimensional data with complex interrelations. Scientific data is not only heterogeneous;
rather it contains different information types. They vary from aggregated data describing
population and diseases (epidemiology, clinical practice, clinical trials), to more granular
patient data and pathological descriptions (health record, clinical history, physical exams)
and to cellular and molecular data (histology, genetic test results and genomic data) [9-11]. Data is thus, traditionally stored in flat files as more efficient data storage and
retrieval systems like RDBMS that are used to handle normalized business data becomes
inadequate if not useless.

Thirdly, researchers increasingly work as teams, often spread all over the world and need
to collaborate to perform their jobs and generate knowledge. There are interrelated
problems that organizations and researchers routinely struggle with in research and
knowledge generation tasks. These are managing dispersed multidisciplinary teams of
workers and improving researcher productivity with the increasing information overload.
The problem that researchers world over face is figuring out what information is valuable and relevant to the task at hand and long term research. How to share this information with the collaborators and fellow researchers without losing ownership of the data? And more importantly, what information to look for and where? There are few tools, other than brute effort in keeping up with this information and helping the scientist relate and apply information to their tasks and understanding. Information tools beyond simple document management and text search are needed to help to collect information from data, do exploratory searches, and find relationships and connect between information sources to create new knowledge and insight.

Fifth, data management systems in experimental labs needs to very flexible so as to able to integrate with new systems, instruments, analysis tools quickly and seamlessly. Scientists cannot wait for the next version of the management software to use better analysis tool that might prove critical in the research.

Lastly, scientific data management in its current form is a relatively new discipline. Businesses had more time to learn from their mistakes and evolve. Scientists have now realized the need to move away from instrument-centric approach to information-centric approach. It would need some time to try out various approaches and evolve standard solutions.
In the nutshell, research labs are increasingly information intensive these days putting increasing demands on researchers and technologies. All the above issues are equally true for business enterprises as well. Business people have built sophisticated systems to manage business data and processes over the years. Scientists, on the other hand, tend to think that the scope of data management is limited to the mere physical data storage and access mechanism. They have now begun to realize that the problem of effective data management cannot be tackled unless the scope is widen to include the meaning of data, its flow and processes that create, share and use data. Discovering and creating new and relevant knowledge is the key to innovation and competitiveness.

1.3 What is Needed: the Big Picture

The prevalent approach to manage scientific data, as we described earlier is proving very ineffective to deal with the new challenges being faced by the scientific community. We have tried to capture the scientific data management approach that is currently being followed in many laboratories in Figure 1.
This approach primarily relies on manually moving experimental data from instruments to local machines with no or minimal centralization of information. Data sharing is therefore achieved by moving data between local machines. Management of data depends on the individual user and as a result searching for information over a collection of data is tedious and often not very effective. Data is analyzed using various tools one by one with minimal pipelining support. It is almost impossible to track back the workflow information of data as this information in some cases is not stored at all and in other cases is not logically associated with data. This approach is still being followed as there are no good data management systems for specific scientific domains that can be tailored to
specific user needs. Also, these systems are unable to keep pace with the emerging tools and management needs of the scientific community.

A good scientific data management system should not only overcome the shortcomings of the manual copy-paste management by automating most of the operations but should also transform the data management lifecycle from as shown in Figure 1 to as described in Figure 2.
To achieve this data lifecycle, the desired scientific data management system should fulfill are a number of requirements. We will discuss some of the major ones in the next section.

Figure 2: Scientific Data Lifecycle - Desired
1.4 Issue in Data Management

Figure 3: issues in scientific data management
1.4.1 Uniform Access to Data and Interoperability Support

Data is often present in many heterogeneous data sources. These data sources might have different access mechanisms and protocols for authentication and access data. The end user, who needs to work on the data stored by various different types of data sources, should not be concerned about access mechanisms or internal physical storage details of the data source. A user should be able to access and work on all the required data files through a single point of uniform access without worrying about the technical details. And this integration should be completely seamless and transparent to the end user.

1.4.2 Providing Meaning to Data

Sharing data without the context might cause problems while interpreting its meaning. The person creating the data might assume certain things about the data that the user is unaware of. For example, price of a material might just be entered as “23.95” with an underlying assumption that the currency is USD without stating that assumption anywhere or storing the currency with price. But a person accessing this data in Europe might get confused about the currency or worse might assume it as Euro. This is called the naked-data problem. So along with data it becomes essential to specify its meaning to make it sharable information.
1.4.3 Data Consolidation and Integration

There might be disagreements in data sets while we integrate information from different data sources and try to present it in consolidated form. Also, there could be cases where some data values are missing or are unknown. These issues might become very tricky to handle at the integration time. This issue of indeterminism of data might be resolved by assigning different trust levels to different data sources or data owners and by storing probabilistic values of data along with the actual values.

1.4.4 Data Ownership and Security

Data stored in different repositories is typically owned by different stakeholders. This data can also be shared with other users with restricted or full access rights. A data management system should honor the data ownership and access rights while providing uniform access to users. It should also provide better mechanisms for sharing data and for enforcing more effective and stringent access control.

1.4.5 Knowledge and Information Discovery

The desired system should provide an effective and fast searching mechanism across different data sources. User should be able to make intelligent queries not only on the header attributes (as project name, file type, owner, date of creation, last modification date, analysis date, instrument etc.) and content of data but also on the semantics (or
meaning) of data and its meaningful relationships with other data. The metadata should be rich enough so as to enable the discovery of new results and associations in the existing data. End user should also be given enough control to associate searchable metadata with experimental/analysis data.

### 1.4.6 Pipelining and Provenance

Scientists often need to perform various operations on data like data analysis. There are workflows that need to be followed to process a certain kind of data for a certain goal, from getting the raw data from an instrument to getting publishable results. Not only should this workflow be automated, but we also need to store the lifecycle information (i.e. provenance) regarding the transformations and operations that experimental data went through to make more sense out of that data. As this information should be integrated with other metadata associated with the data, it makes sense to have a pipelining engine that is integrated with the data management system.

### 1.4.7 Extensibility and Flexibility

A single solution that fits the needs of every scientific domain is very hard to build. But we can build a system which is very simple and powerful at its core so as to support most of the possible interactions by users from various domains and be extensible so as to enable the ability to tailor the systems according to the specific requirements of various communities.
1.4.8 Providing an Intuitive User Interface (Folksonomies)

Usability is more often the deciding factor in users choosing to use, not use or misuse a software system. It’s also more often the most neglected aspect as far as enterprise systems are concerned. The user interface should be intuitive to the user and easy to use. Instead of offering every action possible cluttered on the UI, it should provide the most useful ones visible and upfront. Advanced functionality can be hidden in the menus and sub-menus for power users. Also, the user interface for such a system should be user friendly and not restraining.
Chapter 2: Related Research

2.1 Data Management Systems

Systems used for managing data and performing various operations on data are called Data Management Systems such. Data management systems provide unifying mechanisms for naming, storing, organizing, accessing, searching, and manipulating information objects. Each of the types of data management system approaches focuses on a different aspect, and provides specific mechanisms for data and metadata manipulation [12].

Database management systems are very hard to categorize. Data management systems space is very fragmented as there are too many systems and approaches specific to the needs of different communities. Also, many of the system categories such as Relational Database Management Systems (RDBMS) follow a specific approach but can be used in variety of industry domains. Other data management systems in categories such as Laboratory Information Management Systems (LIMS) are usually made for a specific domain and goal in mind but may follow many different approaches. Hence, it is very difficult to categorize them based on the domain or the approach followed. That is
probably the reason why many people have not tried to do it. Moore et al proposes the following six categories [12]:

- **Data ingestion systems** are primarily concerned with the real-time Observatories, Applications, and Data management Network (ROADNet) project that manages ingestion of data from real-time from sensors. The data is assembled both synchronously and asynchronously from multiple networks into an object ring buffer (ORB), where it is then registered into a data grid.

- **Data collection creation environments** assemble data collections that represent the information objects within their domain. Each collection is registered into a logical name space to provide a uniform naming convention, and standard metadata attributes are used to describe each information objects.

- **Data sharing environments based on data grids.** While collections are used to organize the content, data grids are used to manage content that is distributed across multiple sites. The data grid technology provides the logical name space for registering files, the inter-realm authentication mechanisms, the latency management mechanisms, and support for high-speed parallel data transfers.

- **Digital libraries for publication of data** are created from observational data collections, or from simulations, and are housed within the digital library.
Curative methods are applied to assure data quality. The organizations participating in collaboration may share their digital entities using a collective logical view. These logical views may be based on different taxonomies or business rules that help in the categorization of the data.

- **Persistent archives for data preservation** are mainly concerned with the preservation of the authenticity of the data, expressed as an archival context associated with each digital entity, and the management of technology evolution. As new more cost effective storage repositories become available, and as new encoding formats appear for data and metadata, the archival collection is migrated to the new standard. This requires the ability to make replicas of data on new platforms, provide an access abstraction to support new access mechanisms that appear over time, and migrate digital entities to new encoding formats or emulate old presentation applications.

- **Data processing pipelines systems** are all about fulfilling data automation needs of the organizations. The organization of collections, registration of data into a data grid, curation of data for ingestion into a digital library, and preservation of data through application of archival processes, all need the ability to apply data processing pipelines. The application of processes is a fundamental operation needed to automate data management tasks. Scientific disciplines also apply data
processing pipelines to convert sensor data to a standard representation, apply calibrations, and create derived data products.

When it comes to the scientific domain, roughly three directions dominate existing general data management approaches [13]:

- Scientific data libraries
- Standalone metadata management systems
- Data Grids

Some projects or implementation may involve aspects of more than one approach. Before coming up with any knowledge management architecture, one needs to characterize these directions one by one. We would also explore what has been done in each of these categories and what is missing.

2.1.1 Scientific Data Libraries

Scientific data libraries focus on representing and manipulating complex datasets in I/O systems with flexible data models and high-level data structures. Modern scientific data libraries, however, define data structures that closely match those of applications, and create self-contained datasets by storing metadata such as user annotations together with the original data. Some popular libraries are specialized to serve a single discipline, while
others are more generic [13]. Common and integrated data makes job of a scientist much easier as many experiments won’t be necessary because the answers will already be online. Data acquisition is being automated and storage in various consortia is growing. The need for specialist curation and the desire to have access has meant that large archives are rarely at single institutions.

Molecular biology, scientist are replacing wet chemistry with lookups in the protein and genome data banks (e.g. to determine the function of a gene or protein). Molecular biology and astronomy are the first areas that have been transformed, but the rest of science is likely to follow.

### 2.1.1.1 National Center for Biotechnology Information

The National Center for Biotechnology Information (NCBI) at the National Institutes of Health was created in 1988 to develop information systems for molecular biology. In addition to maintaining the GenBank nucleic acid sequence database, to which data is submitted by the scientific community, NCBI provides data retrieval systems and computational resources for the analysis of GenBank data as well as a variety of other biological data [14]. NCBI has been charged with creating automated systems for storing and analyzing knowledge about molecular biology, biochemistry, and genetics; facilitating the use of such databases and software by the research and medical community; coordinating efforts to gather biotechnology information both nationally and
internationally; and performing research into advanced methods of computer-based information processing for analyzing the structure and function of biologically important molecules.

2.1.1.2 NetCDF (Network Common Data Form)

The network common data form (NetCDF) was first developed at the Unidata Program Center in Boulder, Colorado in the late 1980s. It originally targeted earth science applications, and later also became widely used among applications in astronomy, climate modeling, and geophysics [13]. NetCDF is a data abstraction for storing and retrieving multidimensional data is described. NetCDF is distributed as a software library that provides a concrete implementation of that abstraction. The implementation provides a machine-independent format for representing scientific data. Together, the abstraction, library, and data format support the creation, access, and sharing of scientific information. NetCDF is useful for supporting objects that contain dissimilar kinds of data in a heterogeneous network environment and for writing application software that does not depend on application-specific formats. Independence from particular machine representations is achieved by using a nonproprietary standard for external data representation [15].
2.1.1.3 Model Data System (MDSplus)

Model Data System (MDSplus) is a tree based, distributed data acquisition system, was developed in collaboration with the ZTH Group at Los Alamos National Lab and the RFX Group at CNR in Padua, Italy. It is currently in use at MIT, RFX in Padua, TCV at EPFL in Lausanne, and KBSI in South Korea. MDSplus is made up of a set of X/motif based tools for data acquisition and display, as well as diagnostic configuration and management. It is based on a hierarchical experiment description which completely describes the data acquisition and analysis tasks and contains the results from these operations. These tools were designed to operate in a distributed, client/server environment with multiple concurrent readers and writers to the data store. While usually used over a Local Area Network, these tools can be used over the Internet to provide access for remote diagnosticians and even machine operators [16].

2.1.2 Standalone Metadata Management Systems

Standalone metadata systems are systems that primarily focus on managing data with the help of metadata within a single organization. Standalone metadata systems support the aggregation of non-heterogeneous record-oriented, item-level metadata from diverse collections. These systems can also be used for performing various other operations on data like normalizing, augmenting, indexing or improving the quality. Standalone metadata management systems typically target a particular set of scientific metadata, store them separately from raw datasets, and often manage them through RDBMS
technologies [13]. An organizational information system might contain various data types and applications that use or produce data in diverse data types. Organizations typically, use different metadata formats to represent various these data types. A metadata management system is responsible not only to integrate these different formats and make sense out of it but is often employed to convert from one format to the other.

2.1.2.1 Metadata Catalog Service (MCS)

Metadata Catalog Service (MCS) is a general metadata catalog service developed mainly at University of Southern California (USC), National Virtual Observatory (NVO), and Earth System Grid (ESG) projects. MCS is a standalone catalog that stores information about logical data items (e.g., files). It also allows users to aggregate the data items into collections MCS provides system-defined as well as user-defined attributes for logical items and collections. These descriptions are encoded in structured ways defined by schema or community standards [13]. MCS provides a mechanism for storing and accessing this descriptive metadata and allows users to query for data items based on desired attributes [17]. MCS may be used for storing and accessing metadata about logical files. A logical file uniquely identifies the content of a file. Among the attributes of logical files are: file creator, creating timestamp, etc.
2.1.2.2 Data Warehouse Systems

The term Data Warehouse was coined by Bill Inmon in 1990, which he defined in the following way: "A warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process" [18]. Ralph Kimball provided a much simpler definition of a data warehouse. As stated in his book, "The Data Warehouse Toolkit" a data warehouse is "a copy of transaction data specifically structured for query and analysis" [19]. Typically, the data warehouse is maintained separately from the organization’s operational databases. Data warehousing is a collection of decision support technologies, aimed at enabling the knowledge worker (executive, manager, analyst etc.) to make better and faster decisions [20]. Data warehouses provide decision support to organizations with the help of analytical databases and online analytical processing (OLAP) tools. OLAP technology can organize data in multidimensional tables called data cubes and provides access to the data warehouse through an interactive GUI [21]. Typical functionalities of an OLAP are creating and processing metadata, performing data transformations, creating data visualizations, coming up with detailed and summary views of information, offering sophisticated analysis and querying capabilities at multiple levels across organizational dimensions.
2.1.2.3 *Laboratory Information Management System (LIMS)*

Laboratory Information Management System (LIMS) are designed specifically for analytical laboratories to manage clinical and scientific research data. A full-featured LIMS will manage various laboratory data from sample login to reporting results. Since the early 1980s, LIMS have evolved from rudimentary notebook replacements to application-specific solutions based on RDBMS in client-server architecture. Technologies like web-enabling, wireless computing, Application Service Provider (ASP), and XML have also been gradually introduced. LIMS are widely used in pharmaceutical and life science applications, and often come as off the shelf commercial products [13].

2.1.3 Data Grids

A data grid is a grid computing system that deals with data, the controlled sharing and management of large amounts of distributed data. The term Grid was coined in the mid-1990s to denote a proposed distributed computing infrastructure for advanced science and engineering [22]. The Grid is a term used to describe the software infrastructure that links multiple computational resources such as people, computers, sensors and data. The term Data Grid has come to denote a network of distributed storage resources, from archival systems, to caches, to databases, that are linked using a logical name space to create global, persistent identifiers. Initially developed for the scientific community as a generalization of cluster computing, grid computing is now gaining much interest in other
areas such as enterprise information systems [23]. Data grids provide mechanisms for the virtualization of data. Data grids make it possible to share files that are distributed across remote storage repositories and even controlled by different administration domains [24].

2.1.3.1 caGrid

An enterprise Grid software infrastructure, called caGrid version, has been developed as the core Grid architecture of the NCI-sponsored cancer Biomedical Informatics Grid (caBIG) program. It is designed to support a wide range of use cases in basic, translational, and clinical research, including 1) discovery, 2) integrated and large-scale data analysis, and 3) coordinated study. The caGrid is built as a Grid software infrastructure and leverages Grid computing technologies and the Web Services Resource Framework standards. It provides a set of core services, toolkits for the development and deployment of new community provided services, and application programming interfaces for building client applications [25].

2.1.3.2 Storage Resource Broker (SRB)

Storage Resource Broker (SRB) was created primarily at the San Diego Supercomputer Center. The SRB manages context (administrative, descriptive, and preservation metadata) about content (digital entities such as files, URLs, SQL command strings, directories). The content may be distributed across multiple types of storage systems across independent administration domains. By separating the context management from
the content management, the SRB easily provides a means for managing, querying, accessing, and preserving data in a distributed data grid framework. Logical name spaces describe storage systems, digital file objects, users, and collections. Context is mapped to the logical name spaces to manage replicas of data, authenticate users, and control access to documents and collections, and audit accesses. The SRB manages the context in a Meta data Catalog (MCAT), organized as a collection hierarchy. The SRB provides facilities to associate user-defined metadata, both free-form attribute based metadata and schema-based metadata at the collection and object level and query them for access and semantic discovery [12].

2.2 Discussion

High-level data models defined in scientific data libraries have simplified the programming interface of I/O operations for scientific applications by providing closely matched data structures and implementing multiple language bindings. The standardization of metadata and data formats greatly helps the development of general purpose data analysis tools [13]. However, a common data model that these libraries propose is most of the time very specifically designed for a specific domain. Also, changing data formats and evolving standards pose a challenge to data libraries as changing a common data format is not at all an easy thing to do. Another issue with standards data formats is that there are too many of them. In addition, sharing model that most data libraries offer is not a very good proposition for many researchers as sharing
data means total visibility. Most of the data sharing libraries lack a good fine grain access 
control mechanism.

As scientific datasets grow larger and more complex, separating metadata from data has 
become a beneficial approach, which enables more efficient, and user-friendly metadata 
representations and queries without intermixing with large binary numeric data they 
describe. But we need tools based on the common format data sets to enable more fine 
grain sharing, efficient querying, pipelining support to name a few.

Standalone metadata management systems understand scientific data management needs, 
and each is dedicated to solutions of one aspect or another. Systems like Laboratory 
Information Management Systems, Product Data Management Systems, and Data 
warehouse Systems etc. are very advanced and quite successful in managing data for 
specific kind of applications in various domains. But these systems as the name of the 
category suggest normally cater to the needs of a single organization or a group of 
organizations that are very tightly associated with each other.

Data grids provide a common infrastructure base upon which multiple types of data 
management environments may be implemented. Data grids provide the mechanisms 
needed to manage distributed data, the tools that simplify automation of data 
management processes, and the logical name spaces needed to assemble collections. 
Many technologies from scientific data libraries and standalone metadata management
systems have been applied to even wider areas of collaborations to construct scientific Data Grids [13]. But data grids too, do not provide a complete end-to-end solution for data management. Data grids are more about sharing and discovering and using the shared data effectively and efficiently. Data grids are not very good at taking care of data management issues at organization or individual level. Also, data grids have to depend on other tools and technologies around it for doing many other tasks like pipelining, data consolidation, caching etc.

A very good, elegant, and complete overview addressing the most above fundamental problems is described by Pacitti et al in “Grid Data Management: Open Problems and News Issues”. The authors discuss a set of open problems and new issues related to Grid data management [23, 26]. To address the requirements of Grid data management using P2P techniques, the main techniques which require further research are: schema management, data source discovery, query processing, load balancing, replication and caching, workflow management, autonomic data management, data security and privacy. Research in some of these areas, in particular, autonomic data management, data security and privacy, is still going on and will require much work before concrete solutions emerge [23]. According to Moore et al, other open issues include identification of the appropriate approaches for federating data grids, and the development of capable data flow processing systems for the management of data manipulation [12].
The diversity of scientific data management approaches has resulted from the complexity of their requirements. Data models and formats differ from one discipline to another, and the metadata of interests vary from one application to another. Taking the earth science community as an example, while the global collaboration ESG is being actively developed, a lightweight management system with simple, limited metadata models like DIMES is still preferred by many. This has suggested an important issue of scientific data management systems when extremely data-intensive and heavily collaborated computations are being emphasized, which is the ability to scale downwards in size and footprint as well as upwards for peta-scale storage. A universal metadata schema and a monolithic management system with uniform access interfaces can provide quality services to widely distributed, possibly interdisciplinary collaborations as demonstrated by the success of existing large community efforts. But a simple, lightweight, and specialized data management systems are still highly demanded among scientists working at the other end of the scale [13].

In the nutshell, although many systems exists in the realm of scientific data management, but no integrating architecture exists that allows us to identify requirements and components common to different systems and hence apply different technologies in a coordinated fashion to a range of data-intensive peta-scale application domains.
Chapter 3: The Metadata Approach

3.1 What is Metadata

Metadata has been around since the first library catalogues were established more than 2,000 years ago. The term ‘metadata’ first appeared in the 1960s but became established in the database community in the 1970s. A casual online search for term “metadata definition” leads us to many resources that define metadata as more or less the same way. It is usually described as “data about data” or “information about data”. Although this is the most common and widely used definition, it is often a source of confusion and is ambiguous. The confusion is caused because this definition does not talk about the purpose of metadata and a clear distinction between data and metadata is source of ambiguity. According to Mark A Bayer et al, well-meaning but often misguided efforts to define what metadata is have blurred the line between data and metadata[27].
The Dublin Core Metadata Initiative (DCMI) provided the first major and systematic attempt at defining an interoperable metadata standard that could be extended with specialized vocabularies. Although DCMI provides definition of various metadata terms and a simple and standardized set of conventions for describing things online in ways that make them easier to find. Where the original intent of the DCMI was to define metadata along the lines of the library’s card catalog (using the kind of information that card catalogs typically contain), follow-on standardization efforts have consistently redefined and expanded the metadata problem instead of solving it [28].
There have been efforts from various research and industrial organizations to refine the basic definition of metadata to make it more purposeful and un-ambiguous. According to U.S. National Information Standards Organization (NISO), "Metadata is structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, or manage an information resource" [29]. Gartner describes metadata much the same way as "Metadata is information that describes various facets of an information asset to improve its usability throughout its life cycle" [27]. Another very useful definition of metadata was provided by Tozer as “the means by which the structure and behavior of data is recorded, controlled, and published across an organization” [30]. These definitions are more useful than the first because these deals with the key aspects of metadata’s purposes instead of either defining it ambiguously or un- necessarily fighting over how data differs from metadata.

3.2 Ontology

According to Wikipedia, ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts. It is used to reason about the properties of that domain, and may be used to define the domain. An ontology may take a variety of forms, but necessarily it will include a vocabulary of terms, and some specification of their meaning [31]. This includes definitions and an indication of how concepts are inter-related which collectively impose a structure on the domain and constrain the possible interpretations of terms. Gruber defines an ontology as ‘the
specification of conceptualizations, used to help programs and humans share knowledge’ [32].

There could be many reasons for using ontology in a system. Here are some of the reasons [33]:

- To share common understanding of the structure of information among people or software agents
- To enable reuse of domain knowledge
- To make domain assumptions explicit
- To separate domain knowledge from the operational knowledge
- To analyze domain knowledge

Ontology is often represented in triplets of subject-relationship-object. Resource Description Framework (RDF), a specification from World Wide Web Consortium (W3C) is normally used to define and represent ontologies.
3.3 Folksonomy

In recent years, tagging systems have become increasingly popular. These systems enable users to add keywords (i.e., “tags”) to Internet resources (e.g., web pages, images, videos) without relying on a controlled vocabulary. Tagging systems have the potential to improve search, spam detection, reputation systems, and personal organization while introducing new modalities of social communication and opportunities for data mining. Based on this observation, the popular tags in social tagging systems have recently been termed folksonomy, a folk taxonomy of important and emerging concepts within the user group. Social tagging systems may afford multiple added benefits. For instance, a shared pool of tagged resources enhances the metadata for all users, potentially distributing the workload for metadata creation amongst many contributors [34].

3.4 Why Metadata

There exist many approaches to resolve different aspect of effective data management. In fact, we as a community have been trying to apply various approaches and techniques to resolve these issues with varying degree of success. We think using different approaches for solving different issues is a good approach. But using a metadata based technique is a better proposition. These are a few reasons behind our thinking:
3.4.1 Metadata is Pervasive

Metadata is pervasive in the enterprise. Every aspect of the data generation, management and usage constantly generates metadata. This metadata is context dependent and may redefine an information asset in its discovery, use and meaning.

Figure 5: Data Lifecycle

A good way to think about metadata is to make a distinction between the intrinsic properties of metadata elements and the purpose of metadata. In this way it becomes clear that each data element can be used in a variety of ways and fulfils more than one purpose.
Therefore all axes about data, from object type and location to computation statistics and history, belong to some metadata category. Examples include [13]:

- Information object attributes such as the name, size, creation time, and location of a file.

- Information object format description such as the numeric representation or layout within a file.

- Information object provenance such as the software and hardware used to generate a file, or more generally the history of a file or a collection of files.

- Information object computation summary such as statistical properties of a file collection.

- Information object annotations such as arbitrary user notes associated with a file.

Metadata not only describes data, it is information about how different data points are related to each other.

An example in the table below illustrates metadata generated throughout the lifecycle of an information object. The colors of rows represent various lifecycle stages as shown in Figure 5. When Sample A.raw is created, metadata that gets generated might include things like, the creator, creation date etc as shown in the first three rows. When Sample A.raw is analyzed using a tool, the next two lines might be generated.
Analysis A1.mzxml

...

...

<protein_A>

    <mol_weight> 20 </mol_weight>

    <ph> 5 </ph>

</protein_A>

...

...

...

Now, when we mine the information that is inside the file Analysis A1.mzxml, we get the next three lines on the table. Access control information of the file can be specified as shown in the next three lines. Finally, when a user attaches a tag, the last line is generated.
### Table 1: data Lifecycle - Metadata

<table>
<thead>
<tr>
<th>Subject</th>
<th>Relationship</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample A.raw</td>
<td>Created by</td>
<td>John Smith</td>
</tr>
<tr>
<td>Sample A.raw</td>
<td>Created on</td>
<td>01/05/09</td>
</tr>
<tr>
<td>Sample A.raw</td>
<td>Instrument Precision</td>
<td>0.5 nm</td>
</tr>
<tr>
<td>Sample A.raw</td>
<td>Generated</td>
<td>Analysis A 1.mzxml</td>
</tr>
<tr>
<td>Analysis A 1.mzxml</td>
<td>Analyzed with</td>
<td>MassMartix</td>
</tr>
<tr>
<td>Analysis A 1.mzxml</td>
<td>Contains</td>
<td>protein_A</td>
</tr>
<tr>
<td>protein_A</td>
<td>ph</td>
<td>5</td>
</tr>
<tr>
<td>protein_A</td>
<td>mol_weight</td>
<td>20</td>
</tr>
<tr>
<td>Analysis A 1.mzxml</td>
<td>Can be read by</td>
<td>Dan Jackson</td>
</tr>
<tr>
<td>Analysis A 1.mzxml</td>
<td>Can be read by</td>
<td>Marry Robins</td>
</tr>
<tr>
<td>Analysis A 1.mzxml</td>
<td>Can be written by</td>
<td>Sam McDonald</td>
</tr>
<tr>
<td>Analysis A 1.mzxml</td>
<td>Tagged</td>
<td>Result</td>
</tr>
</tbody>
</table>

#### 3.4.2 Enables Data Independence

Data independence is a key concept in modern data management systems. Data independence means more flexible and extensible is the system. The separation of data and programs is tricky – one cannot see the data without using a program and most programs are data driven. So, it is paradoxical that the data management community has worked for 40 years to achieve something called data independence – a clear separation of programs from data [8]. Metadata is data that exist in addition to the data. The goal of metadata is makes it easier to manage, find, retrieve, manipulate and access data without depending on the physical storage of the data itself.
Metadata enables two forms of data independence termed as physical data independence and logical data independence. These terms are similar to what used for the Database Management Systems (DBMS).

### 3.4.2.1 Physical Data Independence

The goal of physical data independence is to be able to change the underlying physical data organization without breaking any application programs that logically depend on the original data. This is achieved by metadata. As by definition, metadata is independent from physical details of the data. For example, if we extract and store semantic information from an xml file and store it as ontological metadata externally as in RDF format. Now, if we change the file format from xml to doc, all the programs dependent on the metadata remain unchanged.

### 3.4.2.2 Logical Data Independence

While the goal of physical data independence is used to hide changes in the physical data organizations, logical data independence aims to hide changes in the logical organization of the data. This property becomes very critical in the scientific domain where programs and tools have to deal with heterogeneous data. If we map the semantic metadata of each format to a common standardized ontological format, then even a logical change to a format won’t matter for the tools that depend on a standard ontology.
3.4.3 Unifying Approach

Metadata is the common thread in all the things happening to and around data. Also, metadata generated in by one component can be used by the other component or tool to achieve a more uniform and integrated data management approach. Metadata is frequently described in different forms and stored using different technologies. For example, metadata generated by analysis tools should be available to not only the analyst but also to other researchers. This would enable them to better understand the history of data and thus take better decisions. Metadata standards can be developed to help pass and share metadata between technologies and component to promote consistency, productivity and quality through reuse.

3.5 Categorizing Metadata

As is clear from the example in the last section, information objects generate a lot of metadata throughout their lifecycle. We need to categorize metadata in order to manage and use it effectively. As our system would use metadata as the primary means to achieve the goals of an effective scientific data management system, categorization becomes even more vital to us. We have divided metadata into the following five categories:

- Basic Metadata: Physical attributes, header information
- Core Metadata: Access control, provenance, additional attributes

- Domain-specific Metadata: Ontological representation of data

- Logical Metadata: Common ontological mapping

- User Metadata: User/group specific, annotations, folksonomy

---

**Figure 6: Categorizing Metadata**
The lowest level of metadata would contain the most basic information. At this level we would capture physical metadata that include information about the characteristics of data on physical storage systems such as location, identification attributes and replication information etc. Header attributes of data contains information such as author name, creation date, last modification date, size etc.

At the next higher level, we provide access control to data by attaching metadata information with it in the form of access control lists (ACLs). ACL’s can be associated with an information object in the same way. We can also capture the provenance information (derivation history of data starting from its original sources) using metadata, which is of critical importance in scientific research. Other custom header attributes also fall into this category. These attributes may be defined by the system administrator and used by the users capture additional data information not mined at the basic level.

At the domain specific level, we would represent the contents of the data in terms of a domain specific ontological model to define key data attributes and their semantic relationships. We also need an ontological representation of data in order to make the data representation independent of the data storage format.

Logical metadata would logically map different domain-specific metadata to provide the user with a unified ontological model of metadata. At this level, we would have mapping
rules and a common data model representing semantic relationships between data contents from different domains.

At the highest level, we would have metadata specific to a user/group. This user metadata would allow for better management or searching of the data according to individual needs. We can also provide this metadata in the form of simple text (i.e. annotations or tags) often described as folksonomies.

For the initial implementation, we would primarily focus on the metadata that is shown in red font color in Figure 6. The metadata engine that we have built takes care of adding and managing folksonomies to the data. Also, it manages the relationships between information objects in the form of ontologies.
Uniform Access & Interoperability
Providing Meaning to Data
Data Consolidation and Integration
Extensibility and Flexibility
Data Ownership and Security
Knowledge and Information
Pipelining and Provenance

Basic Metadata

Core Metadata

Domain-Specific Metadata

Logical Metadata

User Metadata

Figure 7: Metadata Levels
Table 2 illustrates the metadata that we generated from our earlier example but assigned to the categories that we just talked about. The colors of the table rows represent various levels of metadata as depicted in Figure 6.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Relationship</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample A.raw</td>
<td>Created by</td>
<td>John Smith</td>
</tr>
<tr>
<td>Sample A.raw</td>
<td>Created on</td>
<td>01/05/09</td>
</tr>
<tr>
<td>Sample A.raw</td>
<td>Instrument Precision</td>
<td>0.5 nm</td>
</tr>
<tr>
<td>Sample A.raw</td>
<td>Generated</td>
<td>Analysis A 1.mzxml</td>
</tr>
<tr>
<td>Analysis A 1.mzxml</td>
<td>Analyzed with</td>
<td>MassMartix</td>
</tr>
<tr>
<td>Analysis A 1.mzxml</td>
<td>Contains</td>
<td>protein_A</td>
</tr>
<tr>
<td>protein_A</td>
<td>ph</td>
<td>5</td>
</tr>
<tr>
<td>Analysis A 1.mzxml</td>
<td>Can be read by</td>
<td>Dan Jackson</td>
</tr>
<tr>
<td>Analysis A 1.mzxml</td>
<td>Can be read by</td>
<td>Marry Robins</td>
</tr>
<tr>
<td>Analysis A 1.mzxml</td>
<td>Can be written by</td>
<td>Sam McDonald</td>
</tr>
<tr>
<td>Analysis A 1.mzxml</td>
<td>Tagged</td>
<td>Result</td>
</tr>
</tbody>
</table>

Table 2: Example Metadata – Categorized

Coming up with a flexible and meaningful metadata model is the key here. Data with descriptive metadata may prove crucial in discovery by browsing (i.e. serendipity), which is of enormous value to the scientific community.
3.6 Applying Metadata

Traditionally, data is stored locally with many local copies of the same data available to different people. Not only does this cause a management nightmare but is also becoming more and more unviable to share and move this data. We need to centralize the data in order to enable better collaboration and sharing. Centralization of data also provides uniform access point and minimizes the movement of data. Instead of copying, the data could be shared and only the analysis/search results need to be moved. All the data cannot be put into a single centralized physical storage as ownership and local control over the data is vital in scientific research. So essentially, we may have many heterogeneous data repositories with no centralized access or common data schema. Centralized data sources are better than local copies but pose the same problem as before. What we need is centralized data access but decentralized control over data. The solution is to build logical data collections that federates and consolidates data from different physical data sources [35]. Everyone that needs access to the data would be authenticated and authorized with this common data collection and would be able to view the data in his/her workspace. Also, data collections would need to authenticate themselves with different physical repositories to gain access to local data. The local resources are effectively encapsulated into a collection service, removing the need for researchers to have user accounts at each site where the data sets are stored [36]. This concept of logical data collections would also enable physical data independence as the end user doesn’t really need to know where the data is actually being stored or what protocol or
technology is being used by the data collection to access data. We can also leverage this

![Logical Collections diagram](image)

Figure 8: Logical Collections

physical data independence to provide value added services such as caching for faster
access and replication for fault tolerance etc.
To manage data effectively with the help of logical data collections, not only do we need to know the contents of the data but also its meaning. We do this by means of providing metadata i.e. data about data. Metadata allows scientists to record information about the creation, transformation, meaning and quality of data items and to query for data items based on these descriptive attributes [17].

The data that is stored should be well described so as to enable the machine interpretability. We should provide meaningful metadata in order to better to organize, share, search, analyze, visualize, and publish data. As described earlier, different type of metadata could be provided at different levels of the system to achieve different objectives.

In addition to provide centralized access to data, logical data collections should be able to consolidate data from different/same sources that might be in different formats. In order to provide a consolidated view of data, we need to correlate heterogeneous data and present it in a common and integrated data model [17]. This common data model should ideally be an industry standard model. But, as there is lack of common data models in most of the scientific domains, we can derive our own common metadata model instead. The idea is to map different ontological models to a common model and hence build ontology of ontology’s. New mappings can be created to support new data formats. This approach would enable us to remove the dependency of our data management system on proprietary data formats.
Chapter 4: Implementation: MetaDB

4.1 Component Architecture

Figure 9 shows the complete high-level component architecture for MetaDB. The components that are functional at this stage are depicted in Figure 10. Figure 9 included all the services and components that MetaDB would use to fulfill the need of a data management solution as identified earlier. Let’s look at its components one by one.
Figure 9: MetaDB - Component Architecture
A fine-grained, simple and powerful API is at heart of MetaDB. We call it the Metadata Engine. This core functionality unit provides APIs for managing meta-data, fetching consolidated logical data sets, updating information objects etc. This component is domain independent, very generic and can be used to manage almost any kind of data, even your songs collection. The core component would interact with different data sources using standard access methods like web services or a ReSTful interface.

The whole framework including the core component is built on extensible Java EE framework and provides enough hooks for adding on domain specific functionality to it in the form of plug-ins. For example, we could have a proteomics plug-in. The domain specific plug-ins would typically also have another logically related part, for the UI. The primary purpose of plug-ins is to create a user specific logical view of data. A songs management plug-ins might interpret “artist” metadata differently at the back-end and could also present it a specific format on the UI.
Domain-specific ontological mapping and integration rules are specified in xml files and are interpreted by a generic Ontology Extraction Component, which is a part of the middle tier core. This component, would periodically extract information from different information objects present in separate data sources, consolidate that information based
on the domain mappings and store it in the RDBMS as meta-data for a frequent, faster centralized access.

The Access Control component is independent from the other business logic component as access control is a cross cutting concern. Any client that wants to use MetaDB services including its own UI would have to access it by first having to authorize and authenticate themselves.

Query Service would provide an advanced query mechanism to the user for executing complex queries. Query interface would have a Structured Query Language (SQL) like syntax with the ability to use domain concepts in forming the queries. Figure 11 shows the dynamic architectural diagram depicting how a typical query would be executed.
Workflow Service or Business Rule Engine Service would be used to interface to an external workflow engine or a business rules engine. Caching service would enable object and information caching at the local level to enable faster access to data.

Figure 11: Dynamic Architecture Flow
The user-interface is a web-based rich-client interface that is very intuitive. Simplicity has been preferred over providing every possible feature on the main window. Moreover, plug-ins can be developed and integrated to extend and customize the base UI functionality.

4.2 Enabling Technologies

The metadata and ontological information associated with data in data collections can be stored in multiple ways. The critical aspect that should be kept in mind is that the metadata model should be as independent of the data model as possible. To represent metadata in a simple, minimally constraining and flexible manner, the most common and standard approach is to use Resource Description Framework (RDF) in the XML file format. The underlying structure of any expression in RDF is a collection of triples, each consisting of a subject, a predicate and an object. But considering the size and amount of data, storing this information in flat files on disk and later running queries on them in real time might prove quite inefficient and thus ineffective. A better approach would be to represent the RDF triplets in a relational database management system instead. This would enable us to utilize the inbuilt parallelism and other features provided by RDBMS suited to handle very large data sets [8]. We can pull the required information from data and store it into the database in the form of an ontology and then provide an efficient querying system to the end users. To extract information from XML data files, ubiquitous
XML parsers or XML's standard query languages like XPath and XQuery can also be used.

Following is the database schema of MetaDB. The colored entities represent the tables that have been implemented as part of the initial release. ObjectRelationship table is used to represent ontological relationships in the subject-relationship-object format. Whereas, Tag table is being used to represent user folksonomies and TagRelationship table to model many-to-many relationship between InformationObject and Tag.
Java/JEE has been chosen over other technologies to enable platform independence and extensibility. We have used Java Swing/Applets at the server side to provide a rich client web based user interface. On the server side, Enterprise Java Beans (EJB) is being used to provide a component based extensible architecture.
4.3 Managing Data Using MetaDB – A Real Example

Let us now see how we can use MetaDB to manage data by using tagging and folksonomies for a scientific research project.

A typical proteomics project on a researcher’s hard drive could be a nightmare in terms of finding the relevant information and management because of its sheer size and number of files. Here is the data from a real project. It contains more than 90,000 files in 19 directories with a total size of over 1.2 GB.
Let us now look at the structure of these files. As is clear from the next picture, data files are embedded in many levels of directories. Different types of files e.g. raw data files, analysis result files and result summary files are scattered all over the place and deep inside the folders.
Figure 14: Example - Project Structure

Let us now try to explore this data by looking into one on the analysis results folder. It contains a large amount of files (more than 1,000).
To solve this problem with MetaDB, we can start by simply dragging and dropping the top-level project folder on MetaDB. MetaDB would copy all these files to the data source, store the parent child relationships and present a tree-like structure in the UI. But this tree structure is very different from what we have on the hard disk. We can now rearrange the tree structure, add tags to it, without worrying about the actual physical file structure. As shown in the picture below a tag is a simple text. A user can either use a tag created earlier in the system or they can create a new tag.
Figure 16: Example - Create Tag

All the tags that are created are shown in the left panel as well as in the Apply Tag combo box in the tool bar at the top.
Figure 17: Example - Add Tag

Now, to view all the files related to Sample A (i.e. all the files that have been tagged Sample A) all we need to do is click on Sample A under tags on the left panel. Files are displayed linearly in a tabular format irrespective of how deeply embedded they are in the package hierarchy.
Figure 18: Example - Multiple Views
Figure 19: Example - Multiple Views 2

We can also select multiple tags on the left panel in order to display all the files that have been tagged with either of the selected tags.
Figure 20: Example - Multiple Tag Selection

There are no limits on how to tag files or folders. We can also tag the objects on the basis of the file type (e.g. raw data, analysis results etc), the project structure (e.g. module 1, sub module a etc), the status of the project or file (e.g. pending, done, urgent etc) or based on any other combination of the above. In this way, users can easily switch the perspective of the data based on what they are doing or what they want to see. This dynamicity is the real power of this approach. The next two screen-shots show one example.
Figure 21: Example - Different Aspects 1
Figure 22: Example - Different Aspects 2
Chapter 5: Conclusion and Future Work

After presetting our approach and building the initial implementation, we can conclude that the use of metadata and folksonomies in data management is a powerful approach for managing data collections. Folksonomies empowers users to organize data the way they want to. Moreover, they also use this user metadata to evolve metadata that is used for better the management of data overall. We also feel that use of metadata in general and ontologies in particular opens up limitless opportunities in this area. Domains like bioinformatics can benefit enormously because of their need to view the same data in multiple ways, need to maintain provenance information and their need for easy and fast extensibility.

Use of metadata is increasing for web and application domain and its full potential is yet to be explored. This thesis is just a small but significant step in this direction. As the next step, we need to deploy this system in a large research laboratory setting and let actual users lay their hands on it. The feedback should be collected and should be used to further extend the system and make changes to the existing functionality. We need to validate the system as well as the framework in an iterative manner in as we add features to the system.
We also need to implement various other components such as access control, caching and query service etc. We should also need to extend and configure the system for various IT infrastructures and platforms. We strongly feel that the right way to do it is by making the core functionalities very generic and extensible and then extending the features by developing appropriate plug-ins for a given environment and needs. Also, development should be done in an iterative way and user feedback should be taken into account after each iteration.

We plan to make the project source code open once the project gains a certain amount of maturity. That would enable the scientific communities to make their own extensions to the software by adding plug-ins.
References


Appendix A: MetaDB – Initial User Stories

A.1 Generating RAW Data

Raw data is generated when user conducts an experiment on an instrument using protein samples. This data is important. So it needs to be saved and analyzed later by the data owner or others in the project team.

A.1.1 Exiting

User specifies the desired location for saving the experiment data in a spread sheet (excel file) on the machine attached to the instrument. User conducts the experiment. Data is saved automatically as raw file at the specified location. Data is copied manually to the individual’s machine using a hard drive.

A.1.2 Desired

User logs in to the machine attached to the instrument using his credentials and performs the experiment. Data is automatically saved to the server preferably with an auto
generated name (may include date of experiment, user initials, project name etc.). Data can now be seen by the owner of from any machine by logging in to the application web site. User doesn’t need to copy the data to his drive at all as he/she can perform all the operation on the data on the server. Although, the functionality of saving a local copy of server data can be provided if desired.

**A.1.3 Notes**

Data size for one experiment ~ 20 mb.

10-12 experiments are performed each day.

**A.2 Blanks and Standards Sample Data**

Multiple Blanks and Standards samples are also run with the actual sample. Each sample (including blank, standard or actual sample) produces its own raw file. Blanks and Standards are critical in proving the accuracy and validity of the results obtained. So we always need to save and keep the information along with the actual experiment result. Further, the order in which these samples were run is also important and should be maintained.
A.2.1 Exiting

Raw files generating from blank and standard samples are just data files. These are no associations among the generated results raw files for one experiment. No order is being maintained. Users are trying to accomplish the cohesiveness manually using variety of individual file/folder naming schemes.

A.2.2 Desired

Blank, standard samples that are run along with the actual protein sample are saved as one logical unit of raw files. These files contain similar meta information and hence can be grouped together logically. Also, order of their samples is maintained along with the timestamp for later references.

A.3 Login and Data Organization

Experiment data is intellectual property of the individual/group. Hence, security of the data is an important issue. Also, data generated is huge and contains multiple files generated by different systems. So, logical and effective organization of data at a centralized repository is a challenge.
A.3.1 Exiting

Data is stored on individual systems hard drive as simple files manually categorized in folder hierarchies. Individuals have devised their personal naming schemes for organizing these files and folders. These schemes usually contain date of analysis/experiment and project name in the name of file/folder. But the data is so huge that all these schemes have proven ineffective. There is no mechanism to share data amongst team members and users prefer to keep the data on their own local hard drive because of security and reliability issues.

A.3.2 Desired

User logs in to the online application and is able to look at the data at the server. As user would be verified and validated, he/she would be able to look at the data that he is allowed to look at. The data would be arranged logically so that it is easy to find desired data in an efficient and effective manner. The interface should be extremely simple, intuitive and easy to use. We can broadly have two ways to do it:

**Structured Tree:** We can have a structured tree based organization model having classic parent-child relationships. The advantage of this approach is that it is a widely used
standard approach. So user would be at home from day 1. The disadvantage is that the structure is rigid and difficult to change. A proposed structure:

```
Project 1
  +----- Sample Group/Unit
    |     +----- Sample 1 (Standard)
    |          +----- Sample 2 (Blank)
    |          +----- Sample 3 (Actual)
    |                    +----- xml file
    |                    +----- mzxml file
    |                    +----- Analysis 1
    |                            +----- MassMatrix Results
    |                            +----- Mascot Results
    |                            +----- Results Comparison
    +----- Analysis 2
  +----- Sample 4

Project 2

Project 3
```

**Ontology based Tagging Meta Model:** In this approach we can attach tags/labels to files/file units. User can categorize and organize data by assigning multiple labels. We can also implement access control easily by assigning a shared label to the data. This is a
simpler approach as we won’t have any relationships among different labels. It is very flexible as we can view/organize same data in multiple ways according to situational needs.

A.3.3 Notes

This user story can be bifurcated.

In case we choose to build on an existing system we actually may not have a choice.

A.4 Adding Notes to Data

User wants to add notes to the data files.

A.4.1 Exiting

There is no provision for notes/metadata in current manual organization of data.

A.4.2 Desired

User should be able to add noted to experimental/analysis data. We can implement it using labels if notes are not very detailed (e.g. ANALYSIS_PENDING).
A.5 Finding Experiment/Analysis Data

Huge amount of data is generated by each user on the daily basis in the form of multiple interrelated files. User typically needs to refer back to these files.

A.5.1 Exiting

As there is to meta-data associated with data files, most people find them by name using windows search functionality. This is a frustrating exercise as this type of search is ineffective most of the time and depends on the naming scheme of the user.

A.5.2 Desired

User should be able to see his/her recent data upfront on the server. Also, he/she should be able to browse though the data easily. Moreover, there should be search functionality that should search files based on parameters such as project name, file type, owner, date of creation, last modification date, analysis date, instrument etc.

A.6 Data Search/Analysis using MassMatrix/Mascot

After we get the raw data from experiment, this data has to be analyzed. MassMatrix or Mascot web applications are typically used to do that.
A.6.1 Exiting

Raw data file is converted to xml/mzxml/mgf file using different applications (stand alone). This file is then analyzed/searched using MassMatrix or Mascot application. Parameters are specified on the basis of which this analysis is performed and results are stored on the server. These results can be seen at the server. But most users prefer to download this data in the form of html files to the local system.

A.6.2 Desired

Manual conversion to xml/mzxml/mgf file should be hidden from the user and should be automated as this conversion is just a technical detail and doesn’t provide any business value to the user. So typically, user would select the raw data file on the server and say analyze using MassMatrix/Mascot. Application would ask for parameters and job would be submitted to the cluster. After analysis is complete status of the data would change and user would be able to view results on the server.

A.6.3 Notes

mzxml is the industry standard format. This file is smaller than xml file.

mgf is the smallest of the three and doesn’t contain some data.
MassMatrix has been developed in house whereas Mascot is commercial software.

OMSA and XTendem are the other software that can be used for analysis.

Analysis may take very long time for some files. We might implement an automatic email notification system at a later stage.

A.7 Comparing Results from MassMatrix/Mascot

There is a need to compare different analysis results obtained using MassMatrix and Mascot of same sample. As the internal files created after analysis by both software’s are pepxml files with almost similar schema, technically we should be able to compare the results.

A.7.1 Exiting

There is not automated way to do it. Users do this manually which is tedious.

A.7.2 Desired

User selects two result files of same sample (obtained from same software or different) and say compare. A new results file is generated showing similarities and differences between the compared results.
A.8 Performing Batch Search/Analysis

Most of the times parameters specified to analyze multiple samples are same (this is true especially when these samples are for the same project). So user should be able to submit multiple raw files with same parameters simultaneously.

A.8.1 Exiting

User is bound to analyze samples one by one and each time same parameter value set is provided. This process requires human intervention regularly and therefore wastes a lot time unnecessarily.

A.8.2 Desired

User selects multiple raw files and says analyze using MassMatrix/Mascot or both. System asks to specify parameters. User enters the parameters values for all the files once and system submits all the jobs to the cluster.
A.9 Seamless Integration of Multiple Systems

A.9.1 Exiting

To perform the whole workflow of conducting an experiment on a sample and analyzing the results, a user has to interact with so many different systems (online and local) to perform various tasks.

A.9.2 Desired

User should typically interact with one server application. This system should be able to interact with all the other systems to perform the tasks required by users. This integration should be completely seamless and transparent to the end user.

A.10 Analysis Pipelines

There is a need to define advance form of batch processing of data in serial or parallel modes using various software’s and based on various predefined rules.

A.10.1 Exiting

This has to be done manually.
A.10.2 Desired

User should be able to define complete workflow for doing various analyses and then use these workflows/pipelines repeatedly.

A.11 Data Archival

There is a need of achieving the old data that is not being currently used.

A.11.1 Exiting

This has to be done manually. No streamline process.

A.11.2 Desired

Should be automated preferably by defining some flexible rules.

A.12 Results conversion to Excel Format

Sometimes analysis data needs to be converted to excel format to prepare various reports.

A.12.1 Exiting

It is being done manually and is really a painful process.
A.12.2 Desired

User selects the results file and says export to xls, specifies some parameters and data are converted to the desired format.

A.13 Conversion of RAW Data Graphics using Template

There is a need to convert raw data graphics to a different format (different font etc.)

A.13.1 Exiting

This currently being done manually and takes quite a bit of effort.

A.13.2 Desired

We should be able to define the format settings once and do the conversion automatically.