

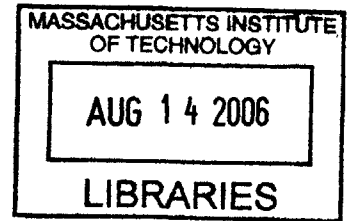
**Designing Customizable End User Applications Using Semantic Technologies to
Improve Information Management**

by

Sumudu Weerakoon Watugala

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

[June 2006]
May 2006



© Sumudu W. Watugala. All rights reserved.

The author hereby grants to MIT permission to reproduce and
distribute publicly paper and electronic copies of this thesis document
in whole or in part.

Author _____
Department of Electrical Engineering and Computer Science
May 2006

Certified by _____
David R. Karger
Professor of Computer Science
Thesis Supervisor

Accepted by _____
Arthur C. Smith
Chairman, Department Committee on Graduate Students

BARKER



Room 14-0551
77 Massachusetts Avenue
Cambridge, MA 02139
Ph: 617.253.2800
Email: docs@mit.edu
<http://libraries.mit.edu/docs>

DISCLAIMER OF QUALITY

Due to the condition of the original material, there are unavoidable flaws in this reproduction. We have made every effort possible to provide you with the best copy available. If you are dissatisfied with this product and find it unusable, please contact Document Services as soon as possible.

Thank you.

The images contained in this document are of the best quality available.

Designing Customizable End User Applications Using Semantic Technologies to Improve Information Management

by

Sumudu Weerakoon Watugala

Submitted to the Department of Electrical Engineering and Computer Science

on May, 2006, in partial fulfillment of the

requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

Abstract:

Personalization capabilities in computer applications attempt to better meet the needs of individuals. The more traditional and widespread paradigm in application design is that the user should adapt to the available application. This requires that the individual user's task be sliced and molded to fit the dimensions offered by an inflexible, monolithic application. It is desirable to have an application that can be shaped to fit each individual user's dynamic needs. However, it is important that this is done in an intuitive and unobtrusive way. In this thesis, we design and evaluate a personalizable application developed to aid life science researchers in their work. We designed the application in Haystack, a platform for developing semantic applications and user interfaces. The application gave the user flexibility in personalizing the way in which information is organized and displayed, while giving users access to the tools necessary to perform their tasks. We selected researchers as the user group to focus on because of the inherent necessity in their work for originality and dynamic adaptation. Life sciences research was chosen as the domain due to its potential to benefit from the application of semantic technologies. We tested how users reacted and adapted to this application by conducting a formal user study.

Thesis Supervisor: David R. Karger

Title: Professor of Computer Science

Acknowledgements

I would like to thank my thesis advisor Professor David Karger for his invaluable support throughout this process. His patience and guidance were imperative in completing this thesis.

I wish to thank Dennis Quan for all his help in introducing me to the Haystack Project.

Many thanks are also due to Karun Bakshi, Dr. Steven Garland, David Huynh, Vineet Sinha, Jaime Teevan, and everyone else in the Haystack Group for their support. Special thanks to Zhenming Liu for his interest and assistance on the project.

I would like to thank Dr. David Kennedy for finding the time for our discussions on the problems faced by life science researchers. I would also like to thank all the participants of the user study for their time and commitment.

I would like to thank anyone not mentioned here who helped me with this project in any way.

Lastly, I thank my family for the countless ways in which they help and support me. I would like to thank my parents and my brothers, Megha and Chula, for their love and belief in me.

Thank you, Amma; you are my strength and inspiration.

Table of Contents

Chapter 1	Introduction.....	17
1.1	Problem Statement.....	21
1.2	Life Science Research - A Case Study	22
1.3	Approach.....	25
1.4	Thesis Overview	26
Chapter 2	Background Work.....	28
2.1	Semantic Web.....	28
2.1.1	Limitations of the Current Web	29
2.1.2	Integration of Semantic Technologies to the Web.....	30
2.1.3	Uniform Resource Identifiers (URIs)	31
2.1.4	Resource Description Framework (RDF)	31
2.1.5	Future of the Semantic Web	33
2.2	Haystack.....	34
2.3	Bioinformatics and Semantic Technologies	35
2.3.1	Unique Naming and Ontology Use.....	36
2.3.2	Literature Search.....	40
2.3.3	Visualization tools	41
2.3.4	Collaboration and Consolidation	42

Chapter 3	Haystack.....	47
3.1	Overview.....	47
3.2	User Environment.....	48
3.2.1	Layout and Starting Points.....	49
3.2.2	Collections.....	50
3.2.3	Views.....	51
3.2.4	Operations.....	52
3.2.5	Workspaces.....	53
3.3	Programming Environment.....	54
3.3.1	Resource Description Framework (RDF).....	55
3.3.2	Adenine.....	57
3.3.3	View Architecture.....	59
3.3.4	Packaging Mechanism.....	60
Chapter 4	System Design.....	61
4.1	Bioinformatics Package.....	62
4.1.1	Database Connectivity.....	62
4.1.2	Analysis Tools.....	63
4.1.3	Data Collections.....	64
4.1.4	Views.....	64
4.1.5	Operations.....	64
4.2	View Editor.....	65

4.3	Other Modifications	66
4.3.1	Feedback Mechanism.....	67
4.4	Challenges.....	68
Chapter 5 Experimental Method.....		71
5.1	Evaluation Framework.....	71
5.2	Test Environment.....	72
5.3	User Study Design	74
5.3.1	Participants.....	76
5.3.2	Information Collections	78
5.3.3	Tasks	81
5.3.4	Establishing Baselines	82
5.3.5	Measures	83
5.4	Overview of an Evaluation Session	84
Chapter 6 Experimental Results		87
6.1	General Observations.....	88
6.2	Analysis.....	90
6.2.1	Main Tasks.....	92
6.2.2	Numerical Data	99
6.3	Design Implications	107
6.3.1	Strengths	108

6.3.2	Limitations	108
6.3.3	Usability Issues	109
Chapter 7	Conclusion	114
7.1	Contributions.....	114
7.2	Future Work	115
7.2.1	Developing the Query Builder	116
7.2.2	Changes to Workspace Layout	118
7.2.3	Extended User Evaluation.....	118
Appendix A	– User Study Overview	120
Appendix B	– User Study Instructions	122
Appendix C	– User Study Feedback Form	139
Appendix D	– User Responses	150
References.....		152

List of Figures

Figure 1-1: A view of an mRNA sequence in BioHaystack showing selected properties, a graphical representation, and possible operations that can be run on it.....	24
Figure 2-1: RDF enables the integration of knowledge from disparate sources of information.....	32
Figure 2-2: An example Life Science Identifier representing a human gene, <urn:lsid:ncbi.nlm.nih.gov:lsid.biopathways.org:genbank_gi:30350027>, has the above properties (among others) to describe and define it.....	38
Figure 2-3: A view of a gene in BioHaystack that displays some of its attributes and its relations to other resources to the user.....	39
Figure 2-4: 3D Slicer Visualizations Facilitate Function BIRN Research	42
Figure 2-5: BioMoby Design.....	43
Figure 2-6: BIRN Site Map.....	44
Figure 2-7: SWAN Prototype	45
Figure 3-1: Haystack start page: the <i>Starting Points</i> collection	49
Figure 3-2: The four main parts of the Haystack layout: the toolbar and left, right and main panes	50
Figure 3-3: Changed view of <i>Starting Points</i> , now shown in the <i>All information</i> view...	51
Figure 3-4: Performing the <i>PubMed Search</i> operation.....	52
Figure 3-5: The editing view of a workspace	53
Figure 3-6: The “Usage” view of the same workspace shown in Figure 3-5	54

Figure 3-7: Examples of RDF Statements	56
Figure 3-8: A statement describing the MIT's homepage	56
Figure 3-9: Different representations of the same set of “triples” describing human hemoglobin	57
Figure 3-10: Describing the MIT homepage in Adenine.....	58
Figure 3-11: Defining a collection in Adenine that holds results from past queries to PubMed.....	58
Figure 3-12: An example method in Adenine – the operation that shuts down Haystack	59
Figure 3-13: Diagram of classes relevant to the view architecture in Haystack.....	59
Figure 4-1: Operations in the <i>Bioinformatics</i> collection.....	65
Figure 4-2: Diagram of classes relevant to the <i>editable</i> view architecture in Haystack...	66
Figure 4-3: Mechanism for sending feedback directly from within Haystack	67
Figure 4-4: Web-based feedback form as an alternative to the direct feedback mechanism added to Haystack	68
Figure 5-1: Haystack User Study Information Page	75
Figure 5-2: Morphometry BIRN frequently uses tools like 3D Slicer for visualizing volumes and surfaces in the human brain	77
Figure 5-3: A view of a data group in Haystack from the hundreds of data groups obtained from IBVD	80
Figure 6-1: A screenshot from a user session: PubMed results obtained for the search “DTI imaging”	90

Figure 6-2: A screenshot from a user session: collection containing records of all past PubMed queries and results	91
Figure 6-3: A screenshot from a user session: the personalized version of the preferred view of a publication.....	91
Figure 6-4: A screenshot from a user session: browsing history	92
Figure 6-5: The average rating given by BIRN researchers	102
Figure 6-6: The average ratings given by MIT graduate student researchers.....	103
Figure 6-7: Comparing group average rating for the task, “Viewing all application windows necessary to complete a task on the screen at once (11g)”.....	104
Figure 6-8: Comparing group average rating for the task, “Querying a database and annotating and sorting the results (12g)”	105
Figure 6-9: Comparing group average rating for the task, “Visualizing data of the same type in different views (13g)”	105
Figure 6-10: Comparing group average rating for the task, “Creating a new way of visualizing data of a certain type (13h)”	106
Figure 6-11: Comparing group average rating for the task, “Making persistent connections between different data sets for future use (14g)”	106
Figure 6-12: Default view with preview pane	110
Figure 6-13: Slide to enter required inputs for an operation in the Haystack right pane	111
Figure 6-14: A context menu, showing submenus for different levels of abstraction....	112
Figure 7-1: Interface to create a “Smart Playlist” in Apple’s iTunes	116

Figure 7-2: Adenine code that specifies a collection data set or dynamic collection in

Haystack..... 117

List of Tables

Table 6-1: Responses for “Annotating and managing your data so that you can organize your thoughts and find important information at a later date”	93
Table 6-2: Responses for “Querying databases related to your research; annotating and organizing results from your search in a way that will be easy to retrieve in the future”	95
Table 6-3: Responses for “Visualizing your information in multiple ways, depending on the task at hand”	96
Table 6-4: Responses for “Defining and exploring relationships between different data sets”	98
Table 6-5: Average ratings and their standard error on tasks for ease of use, usefulness of feature, likelihood of repeating similar tasks, and the estimated frequency of repetition	99

Chapter 1 Introduction

As the number of users of a particular software application rises, it can be expected that the ways in which the application needs to be used also rises. Each user might want to do tasks that are different and do these tasks in different ways. Given such a phenomenon, building monolithic, inflexible software applications that assume every user will perform the same tasks, using the same set of steps may not be as effective as creating personalizable applications that cater to the individual.

This thesis studies the problems in information management that arise due to the differing needs of individuals. We explore the potential benefit to users from personalizable applications, focusing particularly on life science researchers. Specifically, we study how researchers in life sciences need to visualize and manage their information in different ways due to changing objectives.

In this chapter, we give an overview of the whole thesis and present the motivating problem behind it. We introduce life science research as an ideal domain to study the effects of a personalizable application. At the end of this chapter we discuss the approach taken in our attempt to solve the stated problem.

The more traditional and widespread paradigm in application design is that the user should adapt to the available application [1][2][3]. This requires that the individual user's task be sliced and molded to fit the dimensions offered by an inflexible, monolithic application. What if the application is shaped to fit each individual user's continuously changing needs? Moving away from the one-size-fits-all approach, several software ventures are exploring ways in which personalization can be done effectively to set their applications apart from their competitor's [17][18][19]. Personalization elements have been incorporated into many applications, the most ubiquitous examples being customizable features that have been introduced to Microsoft Office and the Windows environment. Web applications such as Google personal home, Amazon, and Windows Live [56][58][59] also focus on giving their customers a personalized experience.

Certain widely used applications provide users with the option of personalizing a set of features in the user interface. For instance, Microsoft Office enables the user to customize toolbars and menus according to what controls he uses most often and what he thinks is the best arrangement. The difference between such an application with features that customize the interface and what we shall call a fully customizable application is that the latter will give the user greater scope in personalizing both the data model and controls, in addition to simply personalizing the interface.

It is important that the personalization features are unobtrusive and seamlessly integrated to the application. The more successful customizable applications generally seem to be

those that are highly usable even without the personalization features. This is because the user would only have to focus on personalizing the application if that is actually necessary for his tasks; even if the user does no personalization, he is still able to perform his tasks as necessary. Unlike with middleware applications, it will not be essential for the user to put in time or effort to initially setup the application; he will be able to use the application properly right away and would only turn to personalization as a natural extension to the user experience.

After exploring existing personalizable applications at greater depth, we designed a system that tries to overcome the shortcomings that we discovered, and still gives users powerful capabilities in terms to customization. We developed the application using Haystack: a platform for developing semantic applications and user interfaces; originally conceived as a *Semantic Web* Browser. The system contained features to personalize the user interface, the data model and controls, and gave the user access to a set of basic tools that were identified as being frequently used by life science researchers, after extensive discussion with, and observation of, researchers in neuroscience.

We evaluated the developed application through a user study to observe how users react to personalization features in applications. The study was designed to test this new paradigm in application design while introducing users to the capabilities of applications built on semantic technologies.

We identified several sets of users to participate in the study from various research groups based on their research field, to test the hypothesis that researchers in particular would benefit greatly from personalizable applications, due to the highly individualized nature of their work. We focused on researchers in fields related to the life sciences because of its suitability for and potential benefit from the application of semantic technologies (Section 2.3).

Two main sets of users participated in this user study: neuroscience researchers and MIT student researchers. The subjects who participated in the user study preferred the personalizable application built on Haystack for tasks which required an individualized approach or tasks that currently do not have a specialized application focused on facilitating it. However, for certain tasks that currently have specialized solutions, users did not indicate a clear preference between the Haystack application and their “current method”, mainly because users were more familiar with their current method and because targeted commercial applications generally provide more functionality that facilitate a narrow set of tasks.

Users were intrigued by the personalization capabilities provided by the Haystack application and expressed their interest in continuing to participate in the evaluation of later versions of the application as they thought it was an idea with a lot of potential to help in their research process.

1.1 Problem Statement

When we think about information management, we have to consider that there are many aspects to how an individual gathers and organizes information. For instance, researchers need to keep track of personal notes and annotations; they need to bring together information from disparate, dispersed locations and each researcher could have individual methods of organizing information that is important to the steady progression of their work.

In terms of visualization, most applications give the user very few options of ways to visualize their information. When talking to researchers in life sciences, it became abundantly clear that most of the time, there is no current way to visualize their data in a way that conveys useful information and promotes understanding. Even if there is a particular tool that allows a researcher to view their information in a certain way, it is rarely customizable: every user of that tool is encouraged to think inside the same box. This is not desirable, especially in research fields, because thinking in novel ways (or outside of the box) and looking at things with new eyes is what promotes new discoveries.

This thesis explores how giving the user tools for personalization can solve these problems in information visualization and management. We study the problems

encountered by past attempts at creating personalizable applications and try to overcome these limitations in our own implementation.

1.2 Life Science Research - A Case Study

The field of biological research is a rapidly expanding area in which many place hope for solutions to problems that pose limitations to man today. Gene technology, stem cell research, drug development, evolutionary biology, ecology, and environmental studies are often discussed in the news, always with eyes to a future with greater possibilities. However, on the road to this future, research in this field faces previously unencountered problems in sharing and managing knowledge.

A major part of research in biology today is concerned with extracting knowledge from abundant and enormous repositories of information. The frequency of new experiments that generate massive amounts of potentially useful data has changed the current research landscape significantly. Many resources have been focused on addressing issues that arise when biologists need effective tools targeted towards data analysis, database management, Web Services use, etc. The need for better methods for collaboration, finding the right information, storing relevant knowledge, visualizing information, etc. has created the field of bioinformatics.

Bioinformatics is concerned with the collection, classification, storage, and analysis of biochemical and biological information using computers. Most efforts in bioinformatics that addressed limitations faced by the life sciences research community concentrated on solving very specific problems in narrow domains and were often neither reusable nor generally extensible. However, there are more and more initiatives that attempt to provide a framework that can be used to solve a broad set of problems [20][21][22][48][52][23].

BioHaystack aims to bring solutions to this field by extending Haystack, a Semantic Web browser with a dynamic and flexible user interface. The initial releases of Haystack and its variants have demonstrated their usefulness in many areas of personal information management such as email, calendar, address book, file management, etc. BioHaystack will allow users to manage *research* information: to access biological data and Web Services, create and save digital experiments, visualize connections between various biological objects through an easy-to-use user interface, etc.

We explored the capabilities of Haystack and how they can be adapted or extended to solve problems faced by researchers in the life sciences. Existing initiatives such as BIRN, LSID, MOBY-S (Section 2.3), and others give good insight into the strengths and weaknesses of the tools that are currently available to researchers. The collaboration between members of the Haystack group and bioinformaticians aided in determining which features are currently used extensively and what further functionality will realistically be useful in an application targeting life science researchers.

BioHaystack gives users flexibility and depth when they explore and analyze data relevant to their own research interests (Figure 1-1). Once the environment is fully developed users will work with interactive and extensible functionalities, in addition to using tools that they already have access to. The user is able to use tools for querying, statistical analysis, Web Services, visualization, and knowledge management from within one unified environment. BioHaystack also provides the capability to view certain kinds of data in a comprehensive manner and it supports queries to research databases such as PubMed and the *Entrez - Life Science Search Engine* (the cross-database search interface used by many biologists) from within Haystack.

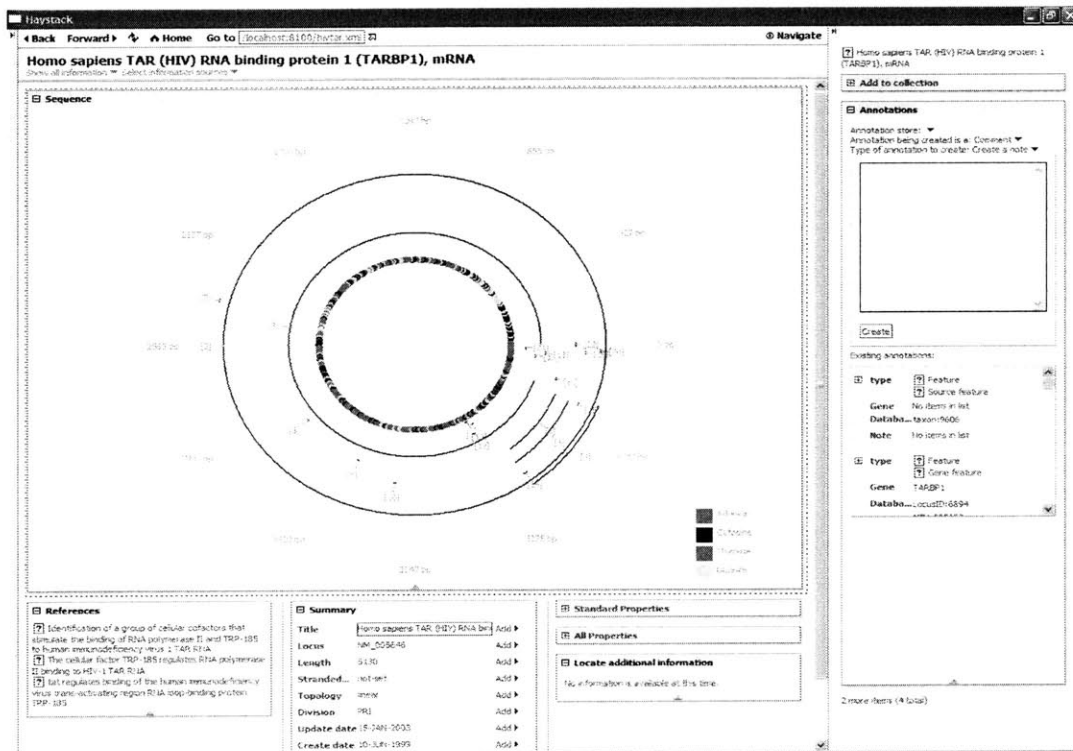


Figure 1-1: A view of an mRNA sequence in BioHaystack showing selected properties, a graphical representation, and possible operations that can be run on it

1.3 Approach

After studying existing personalizable applications and past attempts at applying semantic technologies to bioinformatics, we took into consideration their strengths and weaknesses to determine the best way to solve the problem. We decided on a highly user-centered, iterative approach to the system design: at each stage of the process we presented the application to individuals representative of the target user class to determine their reaction to new paradigms in application and user interface design, and to get their feedback on what features they found particularly useful or would like to see added to the system. We met with users about once a month at the end of each iteration in the design.

Initially, we focused on creating interfaces to bring together data and services so that the user could easily determine connections, correlations, and relevance. This entailed building the user interface to represent related biological objects & biological pathways and getting feedback from researchers in the life sciences to find what forms of visualization helped the most. The final system also includes tools for querying, graphing, Web Services, and other features to improve information management.

For the purpose of evaluating the system, we brought in data on brain volume experiments contained in IBVD [54] into Haystack and created dynamic views for resources of that format. We integrated the LSID [43] resolver, the Bioinformatics view,

and the Entrez Web Service [57] to the Bio-package in Haystack and presented this to actual researchers in different fields of life sciences to introduce them to the new paradigms represented by the system and get their feedback.

By collaborating with bioinformaticians and researchers involved with BIRN and IBVD to determine the focus of the system, we was able to build on the strengths unique to Haystack (such as the existence of a uniform data model, flexibility afforded in terms of information organization, the extensible view architecture, etc.) to create advanced capabilities that demonstrate the benefits of personalization in applications used for research. Through the evaluation of the system, it was also possible to show the benefits to bioinformatics applications from the Semantic Web and related technologies.

1.4 Thesis Overview

This thesis consists of seven chapters. The first chapter introduces the motivation behind the project, the problem that is being studied, and the approach taken to solve it. Chapter 2 presents past work in the domain and related issues. Chapter 3 introduces the Haystack platform on which we designed the system. Chapter 4 presents the design of the application that was evaluated by researchers in the life sciences. Chapter 5 gives a detailed description of the overall design of the user study. Chapter 6 presents the

experimental results and its design implications. The last chapter presents the conclusions: contributions and suggestions for future work.

Chapter 2 Background Work

We studied current tools that allow user personalization in order to better understand which aspects of personalization are the most successful and why. These applications are used in a wide range of domains with different types of target users. We also examined in great detail how semantic technologies are being used in existing attempts to solve problems that life science researchers face in their discovery process. This chapter presents the background relevant to my thesis and discusses prior work on related domains.

2.1 Semantic Web

The Semantic Web is widely seen as the future of the revolutionary technology of the Web. It has been envisioned to provide a framework to enable knowledge sharing without the limitations of the current Web when it comes to interpreting, integrating, and managing information from disparate sources. *Semantic* means to be with meaning; in the context of the Semantic Web, semantics will enable the Web to have *machine-readable* meaning. This would mean that connections between two things (and therefore their relevance to each other) would be well-defined.

Based on over fifty years of research in the Artificial Intelligence community, the Semantic Web began in the World Wide Web Consortium (W3C). The inventor of the World Wide Web, Tim Burners-Lee, sees the Semantic Web as the next logical step in the evolution of the Web [7]: “The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.”

2.1.1 Limitations of the Current Web

The World Wide Web has transformed many aspects of how we obtain and transmit information today. The Web has succeeded in placing information at your fingertips: With a click of a button, resources from around the world can be accessed; it is an equally simple task to publish new content that may be viewed by anyone on the planet. Sharing new knowledge, researching, collaborating, and learning have all been facilitated greatly by the advent of the Web. Existing businesses have been given access to tools that can significantly enhance their customers’ experience; New and highly imaginative businesses have sprung up that take advantage of the vast possibilities of this channel that gives access to a global market.

However, as more people publish and view content, more of its shortcomings have become apparent. It has become difficult to harness the full potential of the Web because of deficiencies in existing searching and browsing capabilities. The Semantic Web was

developed as a solution to the problem of finding the most relevant and reliable content when searching for information on a specific topic.

The Web is a source for an immense amount of information. It is very effective in giving users access to disparate sources of information. But when it comes to interpreting and integrating this information, there is a lot of necessary decision-making that takes time and that must be carried out by a human. Organizing and filtering useful knowledge from the barrage of information with which users get bombarded daily has become a tangible problem.

2.1.2 Integration of Semantic Technologies to the Web

The capabilities of the Semantic Web arise from unique naming and an additional layer of meaning or metadata attached to each name. This means that every object or resource would have a name and a layer of machine-readable information to describe it and specify its connections.

Unique naming and metadata is a way to organize knowledge in a structured manner so that searching and browsing on that framework becomes an essentially automated process. In the Semantic Web, unique naming is achieved through the use of Uniform Resource Identifiers (URIs) and the metadata layer can be provided by using the Resource Description Framework (RDF).

2.1.3 Uniform Resource Identifiers (URIs)

URIs are either Uniform Resource Locaters (commonly known as URLs or web addresses) or Uniform Resource Names (a unique name within a given namespace).

An example of a URI as,

- A URL : <http://www.yahoo.com/>
 - This identifies some information or resource uniquely and also indicates the location of that information
- A URN : <urn:lsid:ncbi.nlm.nih.gov.lsid.i3c.org:pubmed:12441807>
 - A URN identifies a resource uniquely so that there is no ambiguity as to what information object is being referred to. It does not give information about the location: this URN could be located at many different URLs.

In the Semantic Web, a given resource will be uniquely named with a URI. This naming scheme enables programs to assume that a given name will always correspond to the same resource across systems. Therefore, relevant information from dispersed locations can be brought together by identifying connections to a given name.

2.1.4 Resource Description Framework (RDF)

RDF is a flexible and generalized XML format that enables knowledge representation. In the context of the Semantic Web, it is used to describe attributes of resources and

relationships between resources. This is done using “triples” or specifying the subject, predicate and object so that it clarifies the relationship (the predicate) between two resources (the subject and the object).

Figure 2-1 illustrates the principle of RDF that drives the Semantic Web. In this example, information about human hemoglobin is stored in three different locations (GenBank, Gene Ontology and the Protein Database) on the World Wide Web. In the Semantic Web, RDF is used to specify the connection between this molecule and other resources: A text file (containing the gene sequence that codes for this protein), the name of its type and a picture of its 3D structure are all defined to be connected to the human hemoglobin resource by a certain predicate (represented by the named arrow). So if a certain biologist was researching this protein, the Semantic Web will enable him to compile all this relevant information into one preferred view.

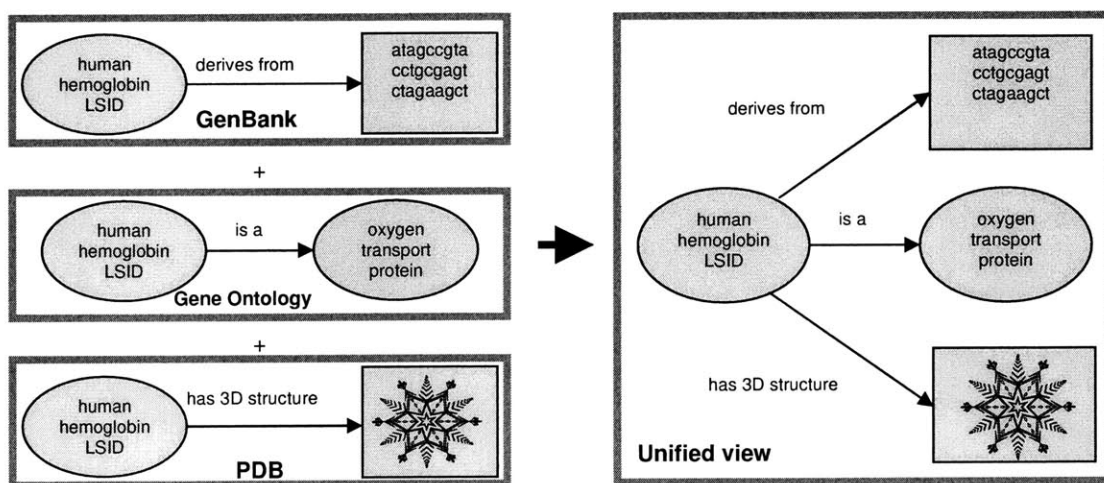


Figure 2-1: RDF enables the integration of knowledge from disparate sources of information

Further details on RDF and how it is used in Haystack is presented in Section 3-3.

2.1.5 Future of the Semantic Web

Consider the following scenario described by Burners-Lee et al. [7] when introducing the vision driving the Semantic Web.

“The entertainment system was belting out the Beatles' "We Can Work It Out" when the phone rang. When Pete answered, his phone turned the sound down by sending a message to all the other local devices that had a volume control. His sister, Lucy, was on the line from the doctor's office: "Mom needs to see a specialist and then has to have a series of physical therapy sessions. Biweekly or something. I'm going to have my agent set up the appointments." Pete immediately agreed to share the chauffeuring.

At the doctor's office, Lucy instructed her Semantic Web agent through her handheld Web browser. The agent promptly retrieved information about Mom's prescribed treatment from the doctor's agent, looked up several lists of providers, and checked for the ones in-plan for Mom's insurance within a 20-mile radius of her home and with a rating of excellent or very good on trusted rating services. It then began trying to find a match between available appointment times (supplied by the agents of individual providers through their Web sites) and Pete's and Lucy's busy schedules.”

As yet, although only a fraction of the Web is currently semantic, the concept of the Semantic Web continues to be adopted in different areas. In fields of research, more and

more resources are becoming semantically described. Incremental steps are being taken towards making scenarios such as the one described above become commonplace in reality.

2.2 Haystack

Haystack is a semantic web browser created by the Haystack Group at MIT CSIAL that has a dynamic and extensible user interface [10]. The Haystack project is driven by the idea that every individual works with information in his own way. In other words, the project contends that personalizable applications are necessary because different users have different needs and preferences.

A variant of the Haystack Semantic Web browser, BioHaystack, was developed to take advantage of the need in bioinformatics to facilitate digital experiments (Section 2.4). The focus of the endeavor is building a version of Haystack that can actually be used by life science researchers for their work. It is built upon the existing capabilities of Haystack integrated with tools that are currently available to life science researchers in order to demonstrate the benefits of using semantic technologies to solve the problems in bioinformatics.

The Haystack Platform is described in more detail in Chapter 3.

2.3 Bioinformatics and Semantic Technologies

There have been a number of recent efforts to bring semantic technologies to bioinformatics. Some of these projects have been quite successful, but their integration into actual research processes has been limited.

Bioinformatics is concerned with the collection, classification, storage, and analysis of biochemical and biological information using computers, especially as applied in molecular genetics and genomics. The need for computerized analysis of biological data has become greater than ever, especially when dealing with the analysis of nucleotide sequences in DNA and other nucleic acids, and when studying the functions, interactions and evolution of genes in various organisms.

A significant part of biological research is done on computers. In Silico experiments or digital experiments have necessitated advanced computing capabilities that enable work with large and disparate data repositories. Bioinformatics tries to address these issues through interfaces that give access to online databases and Web Services. Many endeavors in Bioinformatics attempt to provide researchers in the life sciences with better ways of analyzing and visualizing data. The ease with which users can interact with the huge repositories of potentially useful information could determine how fast the next discovery will be made.

The current trend towards Semantics in bioinformatics stems from the importance, in biological research, of visualizing connections and relationships between various biological objects. The Semantic Web provides an ideal framework for visualizing connections between related information. Many areas in biological research could benefit from such a technology for finding relevance and similarities when dealing with new data, especially since advances in biological fields create a daily flood of new data and information.

Many projects try to make use of this demonstrated advantage from annotations and metadata layers. Groups such as the IBM LSID Project, BioMoby, mBIRN, and ^{my}Grid are exploring the potential of the Semantic Web in bioinformatics. In this section, we describe technologies and tools this project is related to or builds upon, categorized under four types: technologies that enable Unique Naming and Ontology Use, Literature search, Visualization, and Collaboration and Consolidation.

2.3.1 Unique Naming and Ontology Use

There are many initiatives in the life sciences that promote the use of standards in naming and describing biological data and knowledge. This section gives examples of such projects and explains the motivations for this trend of standardizing names and ontologies.

The rate at which new discoveries are made in life sciences can outpace a researcher's ability to keep track of the new information that is relevant to his research. The amount of information and the difficulty to find relevance make this a difficult task. Standard naming and ontologies attempt to ensure that biological information is described in a structured and uniform way so that researchers do not have to deal with disparate data formats. It also simplifies the task of searching biological knowledgebases.

Gene Ontology Identifiers (GO IDs)

The GO Project is part of a larger initiative, Open Biomedical Ontologies (OBO), aimed at classifying biological knowledge using ontologies. GO covers three ontologies: the molecular function of gene products, their role in biological pathways, and their location in cellular components. Gene Ontology supports defining gene products in terms of the ontology and is the most widely used classification standard in bioinformatics. It is used extensively in functional genomics and gene sequencing.

This project has been successful in facilitating research by introducing some standard and structure to the vast amount of data in genetic research. In such a field where new knowledge and data is being added daily, the widespread use of GO IDs has made the work of researchers less tedious and complex.

LSID/BioPathways

Life Sciences Identifiers (LSIDs) are uniform resource names (URNs) for biological objects that are backed by RDF metadata (Figure 2-2). They are an approach to naming and identifying data resources stored in multiple, distributed data stores in a manner that overcomes the limitations of naming schemes in use today.

<i>accession-version</i>	::	<i>BX247883.8</i>
<i>create-date</i>	::	<i>29-JAN-2003</i>
<i>db_xref</i>	::	<i>taxon:9606</i>
<i>division</i>	::	<i>HTG</i>
<i>format</i>	::	<i>urn:lsid:lsid.biopathways.org:formats:fasta</i>
<i>length</i>	::	<i>63983</i>
<i>locus</i>	::	<i>BX247883</i>
<i>lsid_xref</i>	::	<i>urn:lsid:ncbi.nlm.nih.gov.lsid.biopathways.org:genbank:bx247883:8</i>
<i>organism</i>	::	<i>Homo sapiens</i>
<i>primary-accession</i>	::	<i>BX247883</i>
<i>sequence</i>	::	<i>GGCCATCTGCAAACCAATGAGAGAGGCT...</i>
<i>source</i>	::	<i>Homo sapiens (human)</i>
<i>strandedness</i>	::	<i>not-set</i>
<i>taxonomy</i>	::	<i>Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Mammalia; Eutheria; Primates; Catarrhini; Hominidae; Homo</i>
<i>topology</i>	::	<i>linear</i>
<i>type</i>	::	<i>urn:lsid:ncbi.nlm.nih.gov.lsid.biopathways.org:types:dna</i>
<i>urn:lsid:lsid.biopathways.org:types:content</i>		
<i>update-date</i>	::	<i>03-MAY-2003</i>

Figure 2-2: An example Life Science Identifier representing a human gene, urn:lsid:ncbi.nlm.nih.gov.lsid.biopathways.org:genbank_gi:30350027, has the above properties (among others) to describe and define it

The LSID project is a part of the ongoing trend to add semantics to biologically significant resources (individual genes, proteins, and data objects that encode information about them). Many public domain biological databases are available with LSIDs today. For instance, BioPathways.org hosts proxy LSID authorities for several NCBI databases including PubMed, PDB, GenBank, SwissProt, and UniProt.

Within Haystack, the above resource has views that make viewing and understanding LSIDs more intuitive than just looking at the list of properties as in Figure 2-2. Figure 2-3 shows a possible visualization of an LSID (for a DNA object) in Haystack. This particular view shows a list of properties of the gene and displays a “Sequence Summary” prominently, which draws the user’s attention to the most useful information about it. In addition, the view has a node-arrow diagram which shows connections between the current gene and other genes the user is studying.

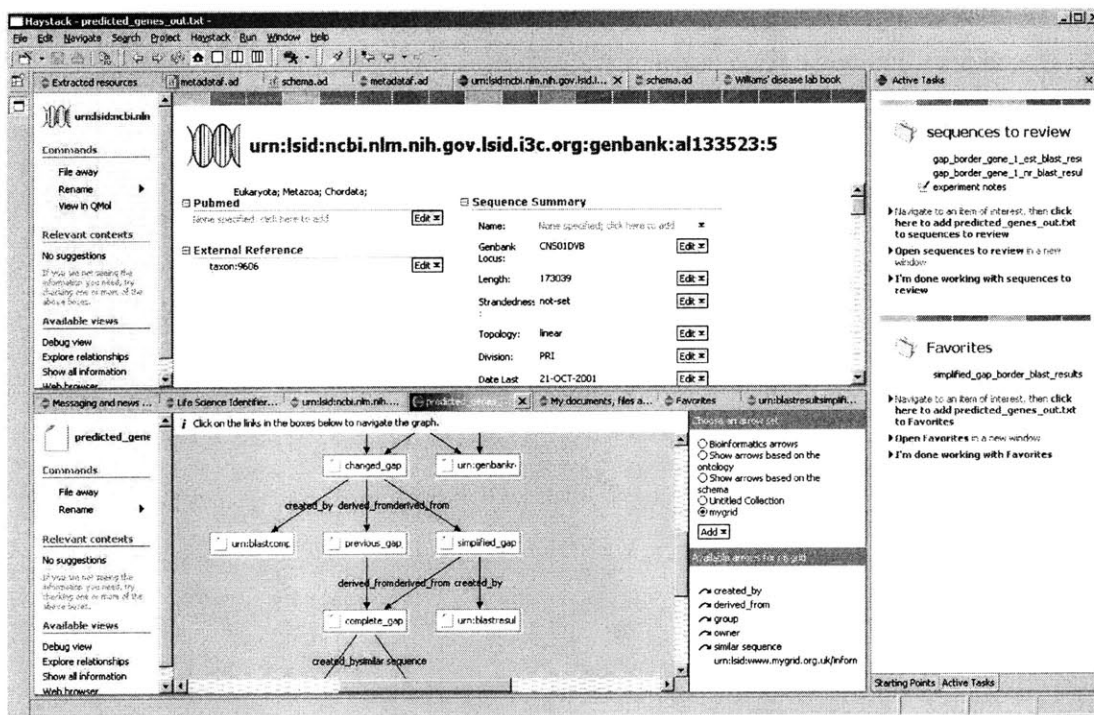


Figure 2-3: A view of a gene in BioHaystack that displays some of its attributes and its relations to other resources to the user

2.3.2 Literature Search

PubMed

PubMed is the most widely used tool to search prior literature in the life sciences. It is a free search engine that is part of the Entrez information retrieval system that lets users search abstracts, references, and citations of publications which are a part of the MEDLINE database. It is also possible to do full-text searches on publications that are available on the PubMed Central Archive.

Various applications such as GOPubMed attempt to merge known identifiers (in this case, Gene Ontology IDs) with results obtained from search queries from PubMed in order to determine similarity, relevance and connections between two biological objects clearer.

BioMed Central

BioMed Central is different from PubMed in that the full-text of all publications in this database is available freely. In addition, the abstract and reference information for all publications are described in RDF. This makes it of particular interest for the Haystack environment.

EndNote

EndNote is a commonly used software tool for publishing and managing bibliographies. Many researchers in the domain use this application to keep track of publications they are interested in and to manage their references while writing papers.

2.3.3 Visualization tools

It is important when dealing with large amounts of data to be able to visualize it in a way that makes sense. Given that life science research generally deals with the analysis of physical objects and systems, it is doubly important to have a useful, efficient visualization capability. 3D Slicer and NeuroLens are just two of many that the life science researchers we collaborated with used in their work.

3D Slicer

3D Slicer is a tool used for the visualization, segmentation and quantification of medical data. Developed through the collaboration between MIT CSAIL and the Surgical Planning Lab at Brigham & Women's Hospital, this software is freely available to researchers. It enables users to visualize data models at different levels and also facilitates “cutting” (Figure 2-4).

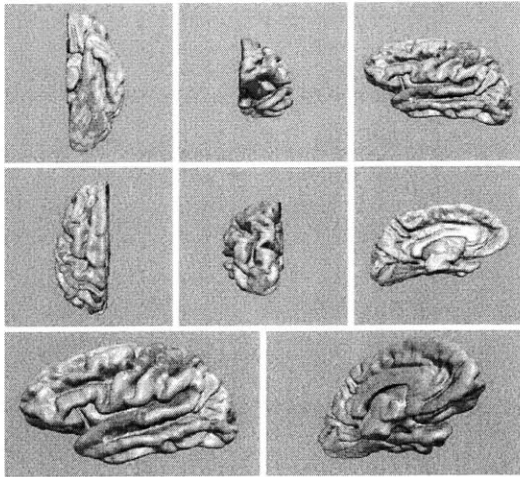


Figure 2-4: 3D Slicer Visualizations Facilitate Function BIRN Research

NeuroLens

NeuroLens is an integrated environment for the analysis and visualization of functional neuroimages [50]. It gives importance to having an appealing user interface that encourages user experimentation and its operations are built on a plug-in model to facilitate extensions to its functionality.

2.3.4 Collaboration and Consolidation

There are many initiatives such as BioPAX, BioPathways, ^{my}grid, Entrez, BioMoby, BIRN, etc. that attempt to develop technology that can be used for collaboration between researchers and the consolidation of knowledge, data, and tools.

S-MOBY (Semantic BioMOBY)

BioMOBY is a central repository of Biological Web Services used by researchers (Figure 2-5). S-MOBY is the Semantic branch of this repository that can be potentially used within Haystack.

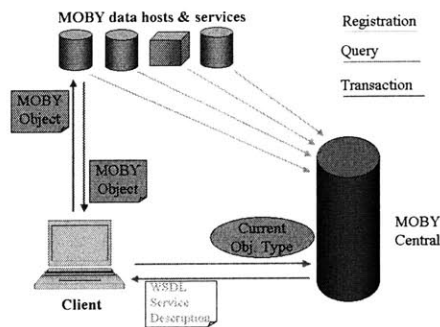


Figure 2-5: BioMoby Design

Biomedical Informatics Research Network (BIRN)

BIRN attempts to address biomedical researchers' need to access and analyze data located at diverse sites throughout the country. It is a National Institutes of Health initiative that fosters distributed collaborations in biomedical science by utilizing information technology innovations.

Currently BIRN involves a consortium of 19 universities and 26 research groups (Fig. 2-6) that participate in one or more of three test bed projects centered on brain imaging of human neurological disorders and associated animal models. Some of these test bed projects are the BIRN Coordinating Center, Function BIRN, Brain Morphometry BIRN, and Mouse BIRN.

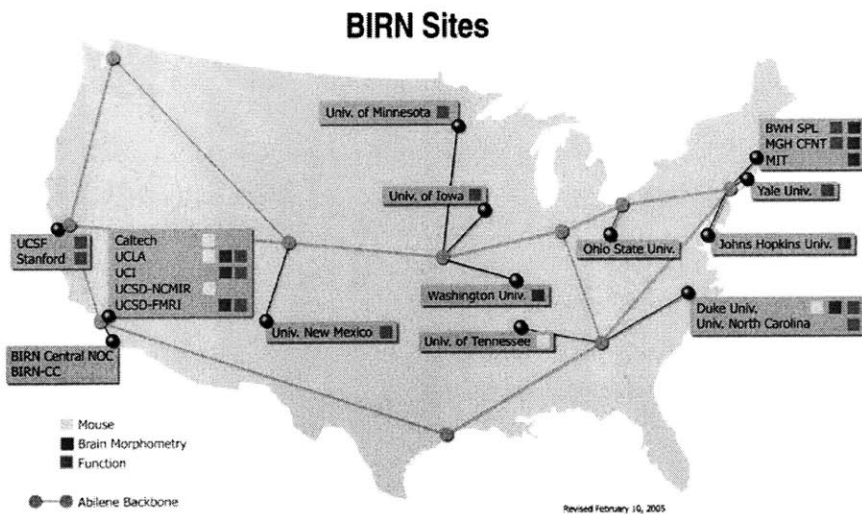


Figure 2-6: BIRN Site Map

BIRN provides a host of services through its network. For instance, it provides powerful tools for visualization and analysis that are used by medical and research professionals who are part of the network.

Semantic Web Application in Neuromedicine (SWAN)

SWAN is a web-based application being [53] “developed as an evolutionary modification of the current scientific knowledge ecosystem”, which brings together scientists, experiments, data, grant applications, publications, scientific databases, bibliographic databases, scientific ontologies, biomedical research collaborations, and scientific web communities, in an environment that promotes collaboration and enhances productivity.

The primary objective to SWAN is to promote and facilitate effective global collaboration between researchers seeking to cure neurological disorders. The current prototype of SWAN (Figure 2-7) attempts to develop effective specialist knowledge bases for the Alzheimer Disease research community, using the energy and self-organization of that community enabled by Semantic Web technology.

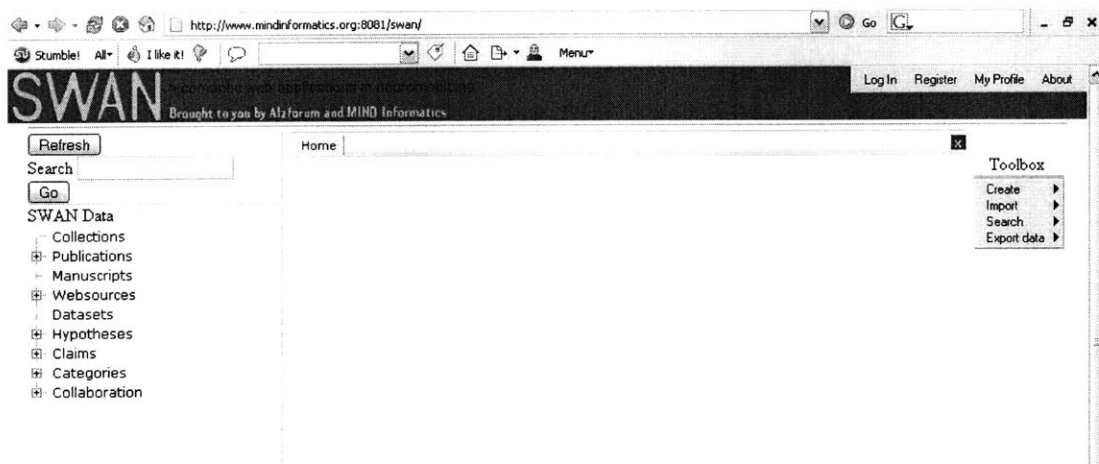


Figure 2-7: SWAN Prototype

SWAN aims to support three principal use cases:

1. Organizing and annotating electronic research materials as integrated knowledgebases across content types, using multiple ontologies, including the ability to create new ontologies,
2. Sharing electronic research materials including ontologies and annotation across different research communities, and
3. Accessing integrated shared electronic research material

Even though SWAN currently focuses on facilitating research in neuromedicine, the concept could be extended to any type of research.

Chapter 3 Haystack

This chapter gives an overview of the Haystack platform and presents its user and programming environments in detail.

3.1 Overview

Haystack can be viewed as a framework that facilitates the development of applications that use semantic technologies. Haystack recognizes that each user could have differing needs as to which information objects should be stored, retrieved, and viewed; what relationships and attributes are worth storing and recording to help find information later; how those relationships and attributes should be presented when inspecting objects and navigating the information space; how information should be gathered into coherent workspaces in order to complete a given task, etc. Haystack attempts to build a better application for information management by building a flexible, extensible, user-centric application.

Driven by the ideas behind the Semantic Web, Haystack aims to let users define their most effective arrangements and connections between views of information (Section 3.2.3) by removing arbitrary application-created barriers, which handle only certain information “types” and relationships as defined by the developer.

Haystack handles all types of information in a single, coherent manner. It provides a single, uniform interface for the manipulation of e-mail, instant messages, addresses, web pages, documents, experimental data, bibliographies, annotations, music, images, etc.

The data types Haystack understands are not hard-wired; any additional types of information that a user wants to work with can be incorporated easily. The user can readily define new object attributes that help him categorize and retrieve information, and new relationships between objects. Rather than being tacked-on afterthoughts, user-defined attributes and relationships are given the same centrality in the interface as built-in relationships such as “author” and “date”.

Haystack attempts to break down barriers created by giving distinct applications responsibility for different data type by matching the information resource in the user’s focus and the operations that can be performed on it.

3.2 User Environment

In this section we discuss the user environment of Haystack. We first present the main application layout and interfaces, and then introduce the resources used in Haystack to organize, manipulate and present information such as collections, views, operations, and workspaces.

3.2.1 Layout and Starting Points

The start page in Haystack presents the user with *Starting Points* (Figure 3-1), which is a collection of resources that the user can navigate to and use for his work.

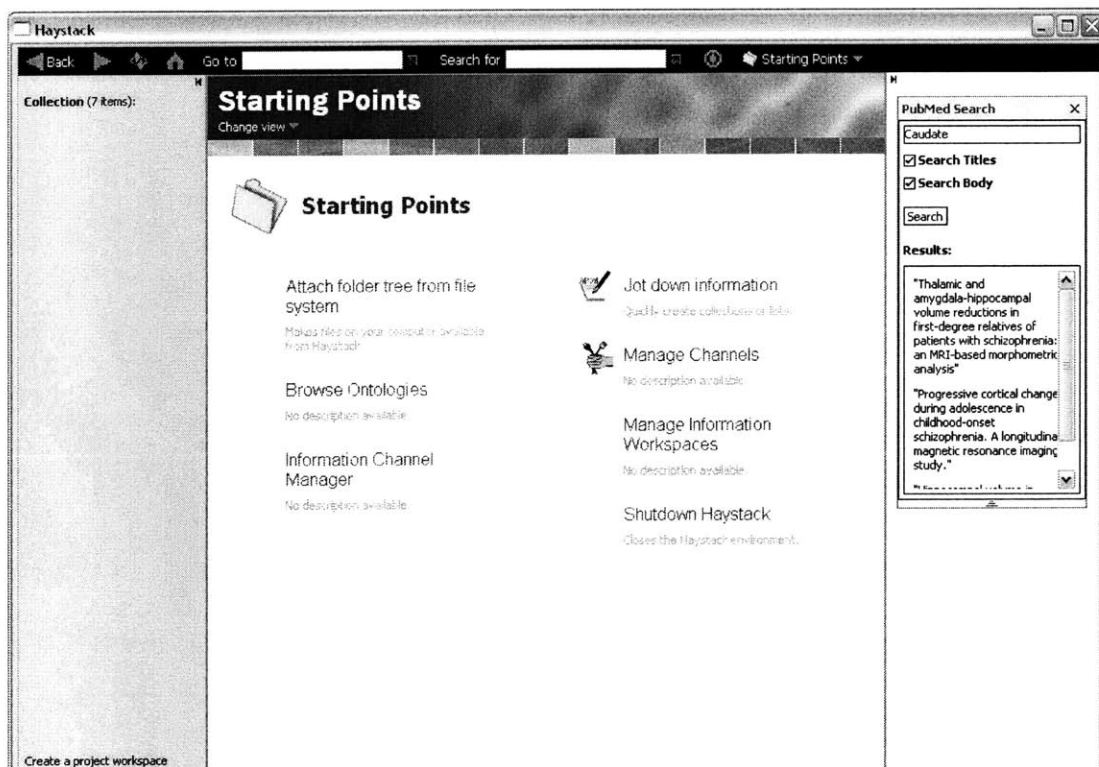


Figure 3-1: Haystack start page: the *Starting Points* collection

There are four main parts to the layout of the Haystack application (Figure 3-2): the menu bars at the top, the main pane, the right pane, and the left pane.

* The menu bar contains buttons for navigation (like in a typical Web browser). This includes Back, Forward, Reload, and Homepage buttons, a navigation box in which the user can specify a URN to navigate to and a search box. Below this top menu bar, there is

a title bar which shows the title of the resource being displayed in the main pane. Just below the title, there is the “Change view” (Section 3.2.3) button.

* The middle pane in Haystack displays the current resource (like an active window in an operating system).

* The left pane changes according to the context, which is determined by what is displayed in the middle pane.

* The right pane can contain anything the user drags on to it (for quick reference), a pane for an operation that the user has just invoked, or a system message.



Figure 3-2: The four main parts of the Haystack layout: the toolbar and left, right and main panes

3.2.2 Collections

A collection is the main resource used for organizing information in the Haystack environment. It is similar to a folder in Windows. However, it is more powerful in that

the user can organize any type of resource in it. For instance, he can add operations, data, views, queries, and other collections into a collection. The *Bioinformatics* collection shown in Figure 3-2 contains other collections such as *My Data*, *Graph Plots*, etc. and the operations *Entrez*, *PubMed Search*, and *Graph Data*.

3.2.3 Views

A view in Haystack is a particular perspective or way to look at an information resource. Figure 3-1 and Figure 3-2 shows the *Browse* view of a collection, which displays icons representing the members of the collection similar to the “Tiles” view of a folder in Windows XP. Figure 3-3 also shows the *Starting points* collection, in the *All Information* view, which displays all RDF triples that have this collection as an object. In both views, the user is looking at the same resource, at different depths of detail and in different perspectives that draw attention to different aspects of the information.

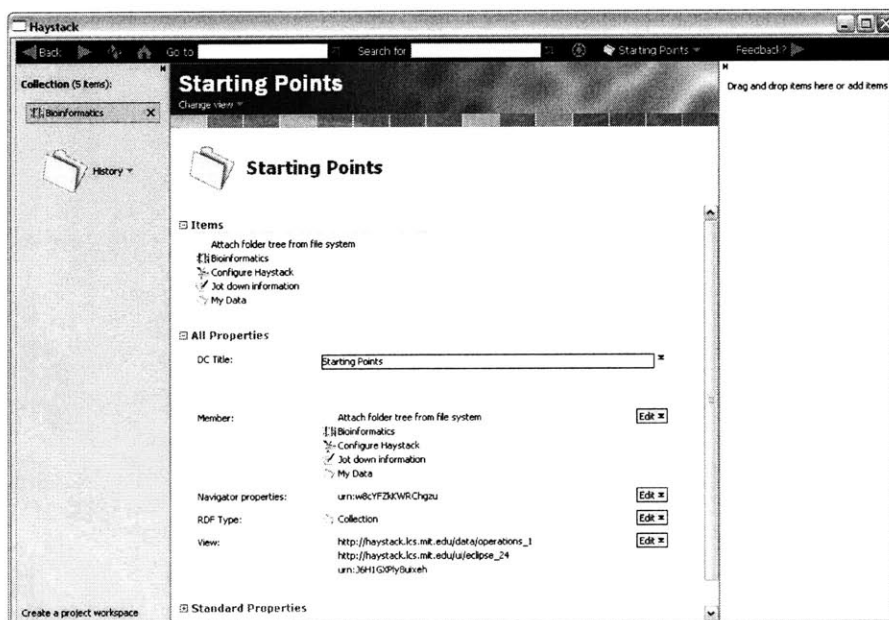


Figure 3-3: Changed view of *Starting Points*, now shown in the *All information* view

3.2.4 Operations

A Haystack operation is a resource that the user can invoke so that some change or addition occurs to data or interfaces in Haystack. An operation (such as spellchecking, statistical analysis, or rotating an image) can be invoked at any time on any object for which the operation “makes sense” (i.e. a blob of text, experimental data, or an image respectively). Operations can also be added to collections in Haystack and like all other objects they can be categorized, annotated, and searched.

Figure 3-4 shows the slide that is presented to the user after he invokes the *Pubmed Search* operation. The slide shows fields for the inputs that the user must give so that the operation can be performed. In this case the user needs to type the *Search String* to be used in the PubMed query. The user can also choose to drag and drop inputs on to the input fields, instead of typing it in. After the input is specified, the user will click the OK button and operation will be performed.

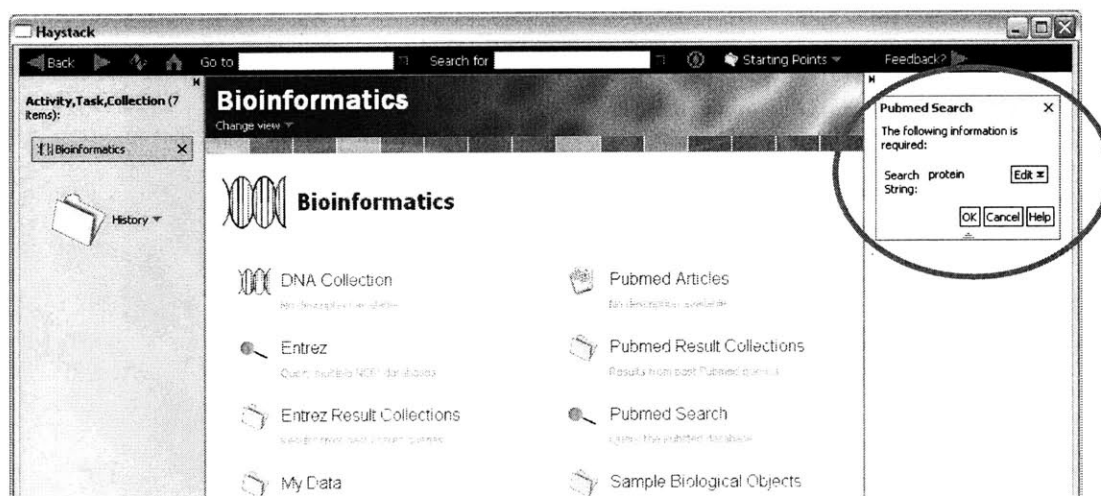


Figure 3-4: Performing the *PubMed Search* operation

3.2.5 Workspaces

A workspace in Haystack is defined to be a collection of panes that the user has put together so that the information necessary to perform a particular task can be viewed on one screen at once [14]. The user can create, edit, and save workspaces according to his needs.

Figure 3-5 shows the editor view of a workspace. This interface gives the user the option of splitting panes to add more panes and lets the user choose which information should be displayed and in what manner.

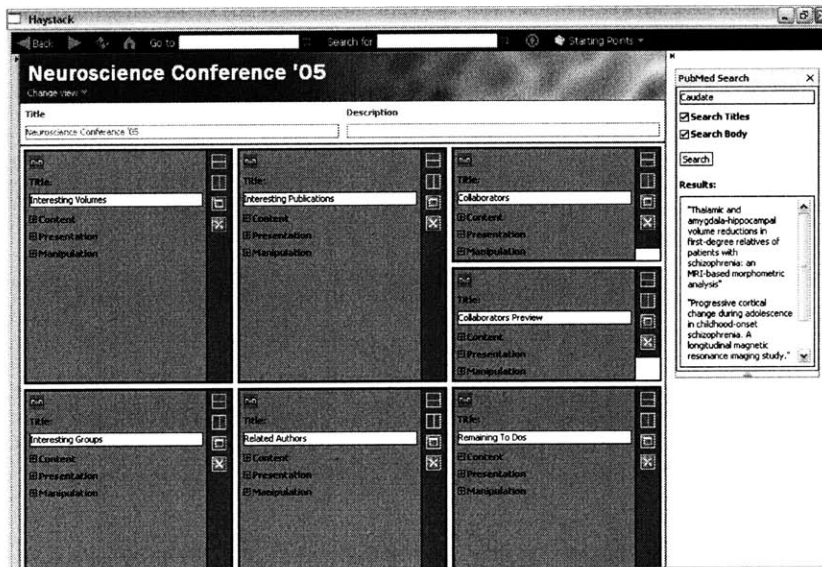


Figure 3-5: The editing view of a workspace

Figure 3-6 shows the view of the same workspace when it is in use. This workspace shows everything a user wants to view in one screen for the task of writing a paper. The

panes show the relevant data, related publications, collaborators who are co-authoring the paper, and the list of remaining things to do.

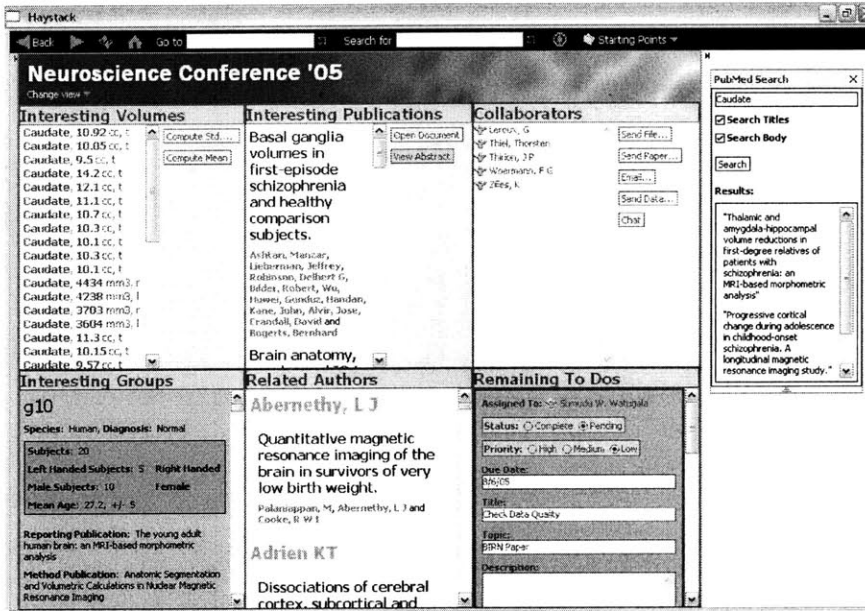


Figure 3-6: The “Usage” view of the same workspace shown in Figure 3-5

3.3 Programming Environment

In this section we discuss the programming environment of Haystack. The section presents RDF and Adenine and describes how they are used in Haystack. At the end of this section, we give an overview of the current packaging mechanism that enables developers to build on Haystack.

3.3.1 Resource Description Framework (RDF)

Haystack builds on a semi-structured data model written in RDF. This section presents the Resources Description Framework as relevant to Haystack.

Using RDF, it is easy to extend the data model to incorporate whatever new attributes or relationships are important to the user. Haystack's RDF database also stores information about which operations can be applied to which resource types and how different types of information should be presented to the user. Storing this information as data instead of hardwiring it in code gives Haystack its ability to adapt flexibly to each user's preferred approach to managing his information. The rest of this section details the fundamental concepts of RDF.

3.3.1.1 Resources and Literals

There are two kinds of entities in RDF: resources and literals.

* A resource is an abstract concept or concrete object of interest. Examples include: John Smith, the act of eating, A Tale of Two Cities, the web site <http://www.yahoo.com/>. Resources are identified by uniform resource identifiers (URIs, Section 2.1.3). By convention, resources will be referred to by enclosing their URIs in angle brackets (<>).

* A literal is a string or fragment of XML. Literals are used to express basic properties of resources, such as names, ages, or anything that requires a human-readable description. Literals are enclosed in double quotation marks (").

3.3.1.2 Statements

Using resources and literals, we can express information in RDF by composing statements. A RDF statement consists of three parts, referred to as a “triple”:

- The subject: the resource being described.
- The predicate: a resource that defines the relationship between the subject and the object.
- The object: a resource or literal that is related to the subject as described by the predicate

Figure 3-7 shows example RDF statements that describe resources within the namespace `urn:examples:` from these statements we can infer that `john` has `hairColor` `black` and `lastName` “`Smith`”, where `john`, `hairColor`, `black`, and `lastName` are resources and “`Smith`” is a literal.

```
<urn:examples:john> <urn:examples:hairColor> <urn:examples:black>  
<urn:examples:john> <urn:examples:lastName> "Smith"
```

Figure 3-7: Examples of RDF Statements

Figure 3-8 shows RDF triples that describe the MIT homepage in Haystack.

```
<http://web.mit.edu/> rdf:type <http://haystack.lcs.mit.edu/schemata/web#WebPage>  
<http://web.mit.edu/> dc:title "Massachusetts Institute of Technology "
```

Figure 3-8: A statement describing the MIT’s homepage

Figure 3-9 shows how information about human hemoglobin could be described using triples. The figure represents the same set of triples as a node-and-arrow diagram and as lines of text. In Haystack, RDF describes information resources, and different methods for viewing these resources can be specified as appropriate.

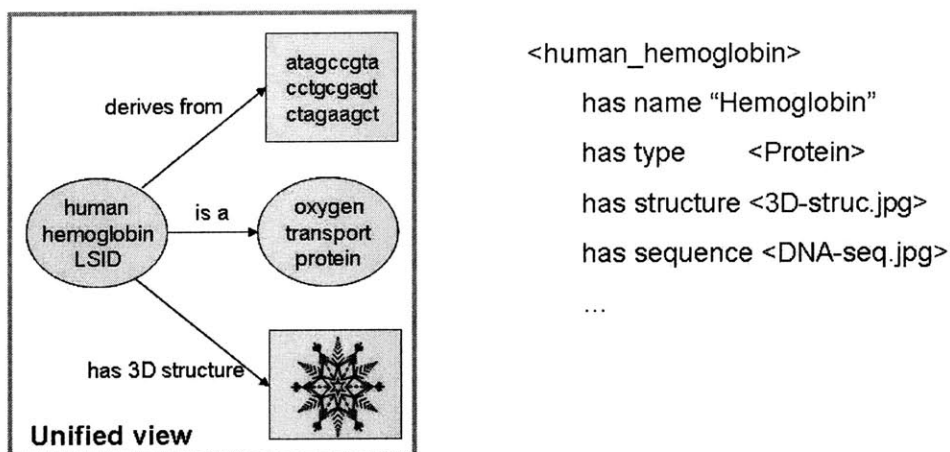


Figure 3-9: Different representations of the same set of “triples” describing human hemoglobin

3.3.2 Adenine

Development in Haystack is done in Adenine [2], a language that was developed especially for this purpose by the Haystack Group at MIT. Adenine is somewhat of a combination between Scheme, N3 and JAVA. It provides a simple way to define RDF triples that can be added directly to the Haystack data model. In the Haystack programming environment every component of the application including user interfaces, methods, operations, functions, data, etc. are described as triples using Adenine.

Figure 3-10 shows how the RDF statements shown in Figure 3-8 above would be added into the Haystack environment using adenine syntax.

```
add {  
<http://web.mit.edu> rdf:type <http://haystack.lcs.mit.edu/schemata/web#WebPage>  
<http://web.mit.edu> dc:title "Massachusetts Institute of Technology "  
}
```

Figure 3-10: Describing the MIT homepage in Adenine

Figure 3-11 shows how to specify a collection in Adenine. Adding this bit of code into a file inside a Haystack package creates a collection titled *PubMed Result Collections* that the user can use to keep the results from past PubMed queries.

```
@prefix pubmed:      <http://haystack.lcs.mit.edu/programs/pubmed#>  
  
add { :pubmedResults  
      rdf:type      hs:Collection ;  
      dc:title      "Pubmed Result Collections" ;  
      dc:description "Results from past Pubmed queries" ;  
}
```

Figure 3-11: Defining a collection in Adenine that holds results from past queries to PubMed

Operations in Haystack are also defined in Adenine and have a metadata layer describing its function that is also specified as triples (Figure 3-12).

```

method          :shutdownOperation ;
adenine:preload "true" ;
rdf:type       op:Operation ;
dc:title       "Shutdown Haystack" ;
ozone:icon     <http://haystack.lcs.mit.edu/data/ozone/common/stopSign.gif> ;
dc:description "Closes the Haystack environment."
               importjava "java.lang" System
               System.exit 10

```

Figure 3-12: An example method in Adenine – the operation that shuts down Haystack

3.3.3 View Architecture

Information resources in Haystack can be viewed in different ways as described in Section 3.2.3. It is possible to specify different views for each *type* of information (possible different views for DNA sequences shown in Figure 1-1 and 2-3, *Browse* and *Table* views for collections, etc.).

The user will have the option of viewing a resource of some type in any view that is specified as a default or possible view for that type of information. Each view is made of several parts that display some set of information about the resource in different ways.

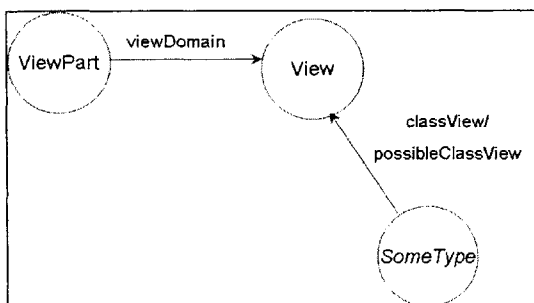


Figure 3-13: Diagram of classes relevant to the view architecture in Haystack

We extended the view architecture in Haystack to handle user-defined “editable” views. This is discussed in detail in Section 4.2 (View Editor).

3.3.4 Packaging Mechanism

Haystack development is done through an open source research project. Over the years, many researchers have worked on adding different functionalities into Haystack such as email, photo, workspace editor, etc. The packaging mechanism attempts to separate out these peripheral projects from the main core Haystack that contains the basic functionality.

Details on how this packaging mechanism was used to build our personalizable application for the life sciences are given in the next chapter under System Design (Chapter 4).

Chapter 4 System Design

This chapter gives an overview of the design of the personalizable application that users evaluated. We present the major components of the system and discuss the design choices that had to be made. Finally, we present the problems we faced in the design and development phases and how we overcame them.

We focused on developing BioHaystack further so that it would contain elements that could be used by real researchers in the life sciences domain to get a good idea about how it would be possible to manage and visualize their research information in this environment. Our end goal was to add a broad set of tools into the system so that the potential of semantic technologies and the Haystack user interface paradigm could be demonstrated properly.

It was necessary to go through an extended period of learning to understand the intricacies of developing applications within the Haystack environment. Some of the challenges we faced during this phase and during development are discussed in Section 2.4.

The final system comprises of several Haystack packages; under the Hayloft packaging mechanism there were ten packages in the application that was tested by users. The main

ones were the Bioinformatics package, the Workspace Designer, and the View Editor. Other packages were the standard Haystack packages together with data and support packages.

The resulting system was highly personalizable but users could use the application for their work without being required to customize or initialize essential components of the system. In addition, even though it is possible through Haystack to give users the capability to personalize the base application layout, this was recognized as unnecessary and error prone, and core elements of the application were made fixed. This meant that personalization was possible up to a reasonable point but the user couldn't modify the application to an extent that made it unusable.

4.1 Bioinformatics Package

4.1.1 Database Connectivity

We added features to query the BioPathways LSID proxy, Entrez (without the LSID proxy), PubMed, and BioMed Central. Every database required a different method of connecting and required different tools and skills: JAVA, XML, Web Services, Adenine, etc.

Information that was received from such databases was time-stamped and organized using collections, depending on the search query that was used to obtain the information, the database it was received from, and the type of the information itself.

When developing this aspect of the bioinformatics package, the inconstancy of outside database APIs was a problem. After implementing a working system to query a certain database, it was not uncommon to see it fail weeks later due to changes in the interface or data formats.

4.1.2 Analysis Tools

Graph plotting tools that enabled users to make simple data plots were added to the system.

Initially, we explored the possibility of adding an extensive JAVA graphing package to Haystack. However, the intricacies involved with this turned out to be too complicated; for the purposes of this project, it was sufficient to add basic plotting functionality since it was only necessary to give users an idea of the different ways in which they could view their information.

4.1.3 Data Collections

It was necessary to add sample data to the system for evaluation purposes (Section 5.3.2). Real data from LSID servers and IBVD was added to the package, and static and dynamic collections (that would update if and when new data was added to the system) were created to keep track of different types of data.

4.1.4 Views

Views were created for certain types of data that was going to be used extensively as part of the user study. Certain existing views required significant modifications due to changes in data formats and due to major changes in the Haystack architecture.

4.1.5 Operations

It was necessary to add Haystack operations in Adenine for each of the databases, Web Services and other tools that were developed for the system. Figure 4-1 shows the *Bioinformatics* collection in Haystack which presents some operations to users that can be used to query life science databases.

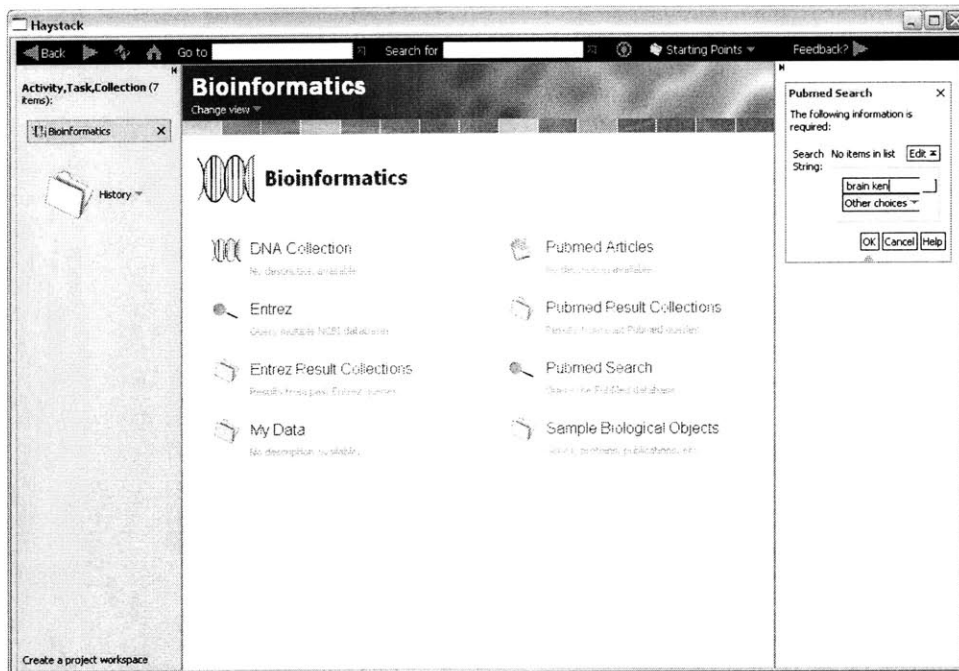


Figure 4-1: Operations in the *Bioinformatics* collection

4.2 View Editor

The view editor enables to the user to create custom views which can be used to display any information resource of a certain type according to the user's preference. Figure 4-2 shows the relevant classes for editable views. The user defined views is both of type *View* and *EditableView*. Once the user selects the "Create View" operation, he is presented with the option of choosing from predefined view-templates. Currently, we have editable view templates that display resources as a table (*EditableTableView*), as a layout of multiple slides (*EditableLensView*), as a list (*EditableListView*), and as a layout of

multiple windows (*EditableWindowsView*). The user can select the number and type of properties to display in these templates.

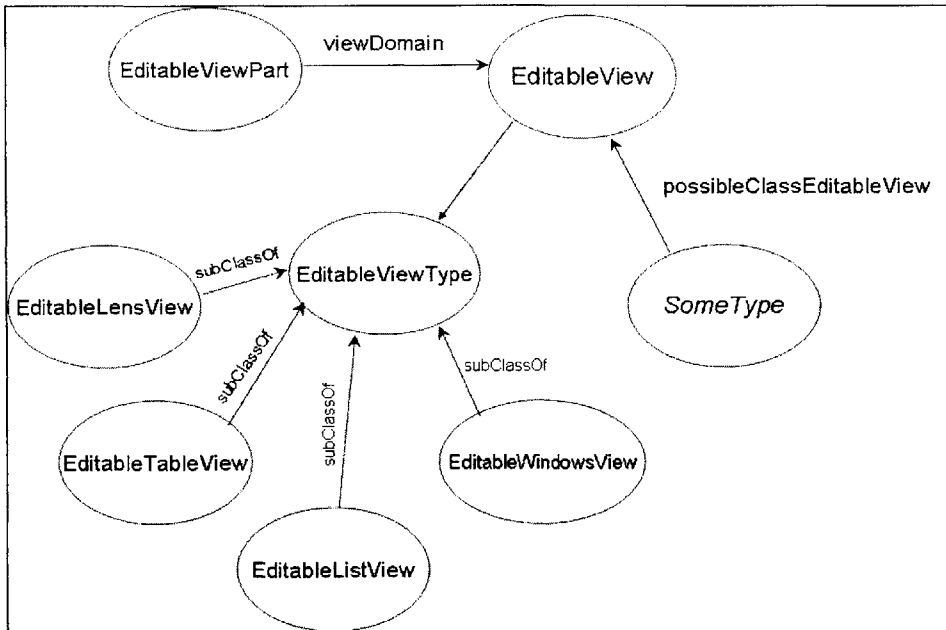


Figure 4-2: Diagram of classes relevant to the *editable* view architecture in Haystack

4.3 Other Modifications

We made various minor changes to different packages in Haystack such as fixing bugs and adding and changing the behavior of a particular feature as appropriate for the Haystack Evaluation Version that was used in the user study.

4.3.1 Feedback Mechanism

We added the functionality to send feedback directly to developers from within Haystack (Figure 4-3). This also included a screen capture mechanism so that it would be easier for the user to show what he was currently encountering.

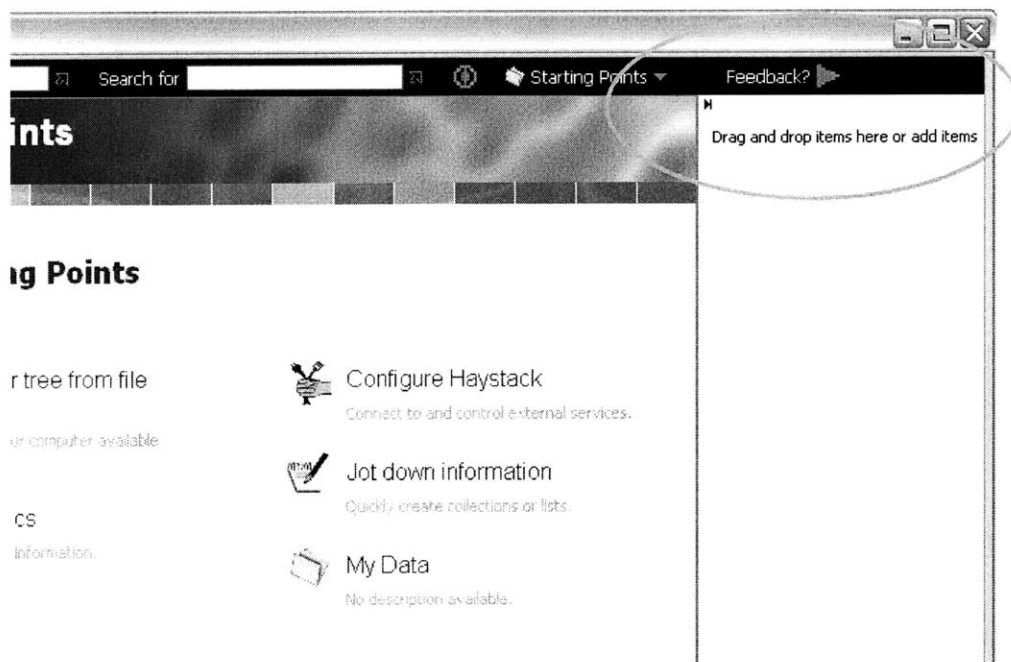


Figure 4-3: Mechanism for sending feedback directly from within Haystack

In case the application was completely frozen or otherwise incapacitated and it wasn't possible to use the direct feedback mechanism properly, users were given the option of using a web form (Figure 4-4) to send feedback.

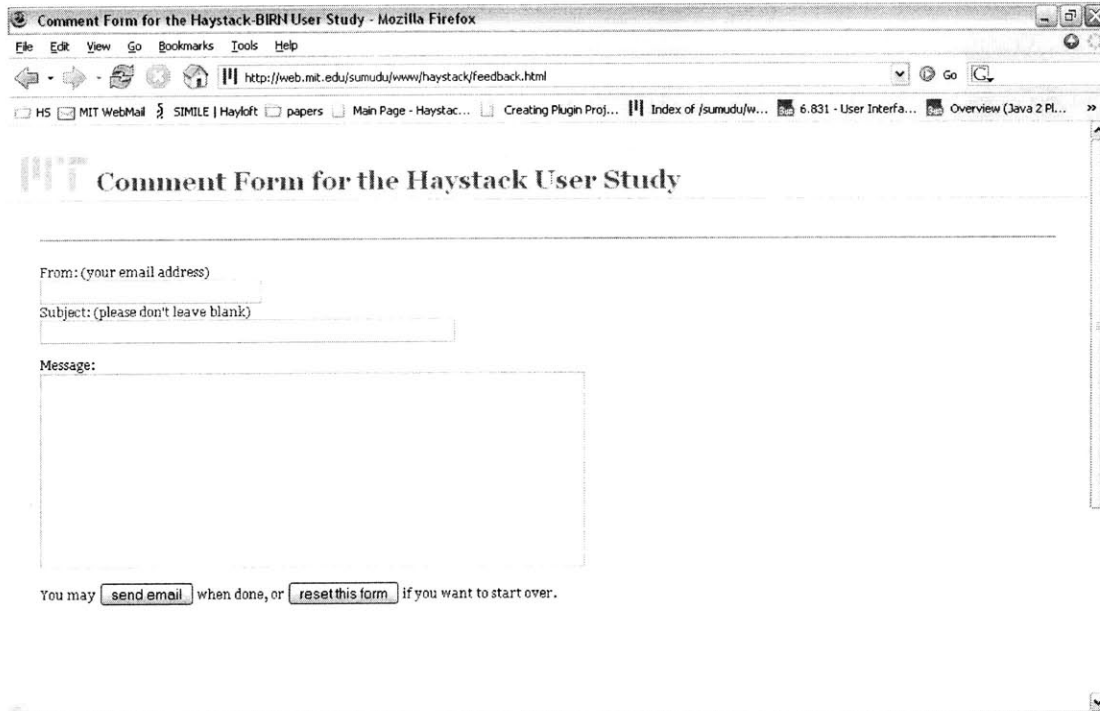


Figure 4-4: Web-based feedback form as an alternative to the direct feedback mechanism added to Haystack

4.4 Challenges

Debugging any errors we encountered while developing on the Haystack platform was a particularly challenging task. As with any large software project, the sheer number of different modules that interact together sometimes made even the most experienced of Haystack developers take pause when presented with some of the issues we encountered during development.

There are several key issues that mask the full potential of Haystack and hinder more widespread development within the environment. In this section, we present some of the problems discouraging potential developers from choosing to work on Haystack to build applications that target information management and visualization.

The lack of complete and up-to-date documentation for current versions of Haystack is a major obstacle to learn about the system quickly and deal with problems when they arise during development. The size of the Haystack code repository makes it harder to understand and visualize interactions, especially when the interfaces between different modules are not well specified.

Different students with different styles of programming have worked on the system over the years. It is imperative that a uniform way of documenting the system is used to keep track of all the different aspects of Haystack. It will be well worth the effort to decide on such a standard as a group and initiate documentation system that will be readable to developers even outside of the Haystack group.

A key element of development within Haystack is the need to use Adenine to specify almost everything in a manner in which Haystack can understand. This almost involves implementing everything twice. For instance, when we implemented querying Entrez, we had to use JAVA classes to directly query the database and then write methods in Adenine files to call these JAVA methods. Then it was necessary to add these files to

bootstrap files of the right package. It may make development within Haystack much easier and accessible if the implementation shift completely to JAVA.

Many different third-party packages are used for different parts of Haystack. In some instances, the interactions between those and packages within Haystack is very complex, particularly in the version of Haystack that uses the Eclipse plug-in package mechanism.

Chapter 5 Experimental Method

In order to determine how users react to personalization features in applications, we designed and ran a user study on the system described in Chapter 4. The study was designed with the aim of getting in-depth, qualitative feedback from individuals from select groups of users. This helped us evaluate how personalization aids individuals in their work by enabling us to observe in detail how different users used and reacted to the system. This chapter presents the evaluation framework, the test environment and the design of the user study that was conducted.

We identified several sets of users to participate in the study from various research groups based on their research field, to test the hypothesis that researchers in particular would benefit greatly from personalizable applications, due to the highly individualized nature of their work. We focused on researchers in fields related to the life sciences because of its suitability for and potential benefit from the application of semantic technologies (Section 2.3).

5.1 Evaluation Framework

Since the system was developed in close consultation with researchers in the life sciences, it was possible to design a framework to evaluate the system based on a good

understanding about what type of tasks users frequently need to do and what resources and tools the user already has access to in the domain. We had extensive discussions with researchers in order to determine the limitations of current methods for performing their tasks.

When designing the Bioinformatics package to Haystack, we included the most commonly used tools such as PubMed, graph plotting, etc. because in order to evaluate how personalization aids users' work, it was necessary that the user be able to perform natural and necessary tasks with the system. In addition, sample data used in the study was also real data that researchers currently use to make new discoveries.

The usability of the system was tested by asking the users to perform tasks that were selected to cover a range of different types of tasks as determined by prior investigation. The user study was designed to be formative in nature; the user was given ample opportunity to give suggestions and feedback to be incorporated into future versions of the system.

5.2 Test Environment

We initially intended that the user study be in-depth case studies on how a set of life science researchers in related domains reacted to our personalizable application. The aim

was to ensure that the test environment be as naturalistic as possible to determine the real benefit of personalization features on the users' research work.

However, it became clear that at the current stage of system development, it was necessary to have more control over the environment than it was possible in a completely natural setting. There were a particular set of tasks that the life science research could perform and the system was not tested to handle errors that pop up with long term use. Therefore, the user study was run in a laboratory setting. However, attempts were made to make the environment, data, and tools that were used by the subjects simulate the real world as closely as possible.

The artificial restrictions were limited to the following:

- The user was requested to do a specific set of previously selected tasks (albeit tasks that were formulated to match tasks which were determined to be frequently performed by the users during prior observation).
- The tasks were performed at a prearranged time unlike in a long-term evaluation where the user would have more flexibility about starting, stopping and coming back to work with the system.
- An observer was present while the user performed the tasks and the user was interviewed by him at the end of the study.

The test environment matched the real world in that:

- The user study was performed in the user's normal workplace, generally on the user's computer.
- The tasks were performed on real data and knowledge bases.
- The tasks were designed to fit into scenarios that were as realistic as possible.

5.3 User Study Design

This section gives an overview of the user study and presents the main ideas behind its design. The subsections describe the participants of the user study, the information collections on which the users performed tasks, the tasks themselves, the baselines of evaluation, and the usability measures that were evaluated. We describe each of these aspects of the user study in greater detail and explain the design choices involved.

The user study comprised of two parts: the first part was designed to help familiarize the user to the Haystack environment; the second part stepped the user through different tasks to demonstrate the range of features in the system and to show how different aspects of the application can be personalized.

The users were first given an overview (Appendix A) of the user study, together with a brief introduction to Haystack. The overview included information about the logistics of

the study like estimated time commitment, recommended system requirements, where to locate resources to get help and send feedback, etc. The users were also given a general idea about the motivation and purpose of the study.

Then the users were given instruction documents (Appendix B) which stepped the users through different tasks designed to help them gain an understanding about how different tasks can be performed with the system, and how features can be personalized according to individual preference.

To allow for users who were going to participate in the study remotely, all the necessary information was published online (Figure 5-1).

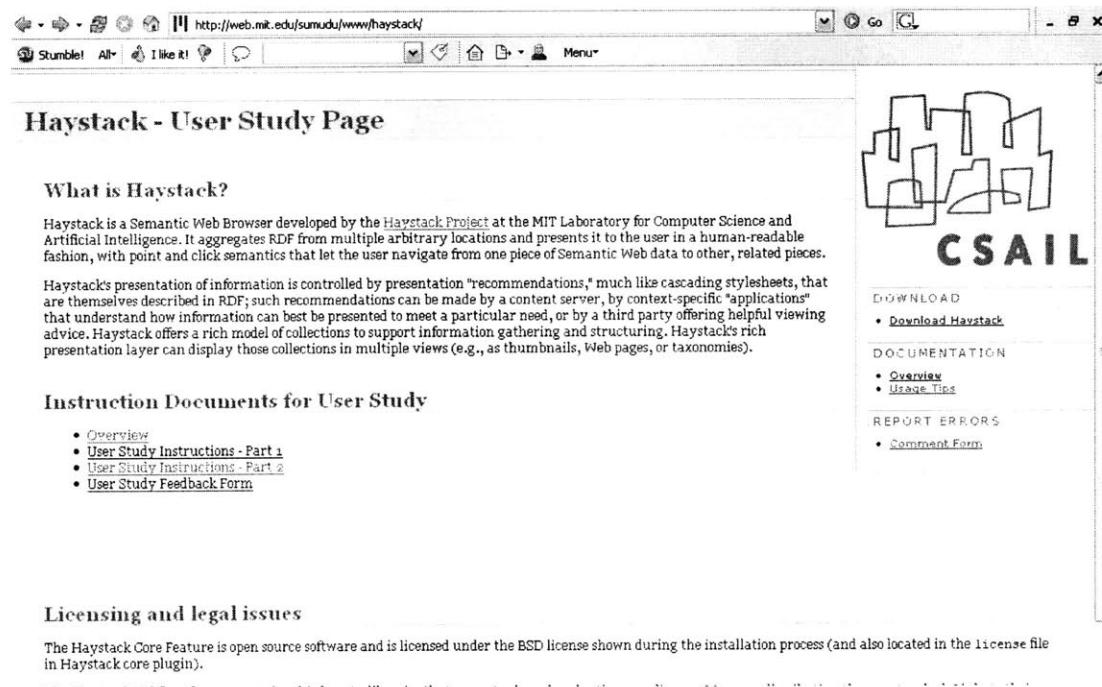


Figure 5-1: Haystack User Study Information Page

Subjects were provided with a variety of avenues to give feedback: direct email access, the Haystack feedback mechanism, an online web form (<http://web.mit.edu/sumudu/www/haystack/feedback.html>), the User Feedback Form (Appendix C), and a one-on-one interview at the end of the study.

5.3.1 Participants

Three user groups participated in the user study: researchers from the Biomedical Informatics Research Network (BIRN, Section 2.4.4.3) at Massachusetts General Hospital (MGH), MIT EECS graduate student researchers with interests in the life sciences and bioinformatics (identified here as the MITLS group), and other students with research interests in user interface design, and information management and visualization. The main user groups were the BIRN and MITLS groups because of their specialized involvement in life science research.

Biomedical Informatics Research Network (BIRN) Researchers

The Haystack group collaborates with Brain Morphometry BIRN (mBIRN) which focuses on studying regional brain dysfunctions related to the progression and treatment of specific diseases such as schizophrenia or Alzheimer's. The stated goal of mBIRN is to “develop capability to analyze and mine data acquired at multiple sites using processing and visualization tools developed at multiple sites” in the context of human

brain MR-based Morphometry. Brain researchers affiliated with mBRIN participated in the user study. Researchers in this group study the structure and volumes of various parts of the brain, and try to find causes and correlations between their findings and various brain disorders and diseases. Like most researchers in the life sciences, they need to explore existing knowledge bases (usually found in different locations and/or disparate formats) for their work.

Different methods of visualization using MRIs, diagrams depicting different layers of abstraction/scale of brain images (Figure 5-2), etc. are particularly important for this group for viewing brain structures and their relations to one another

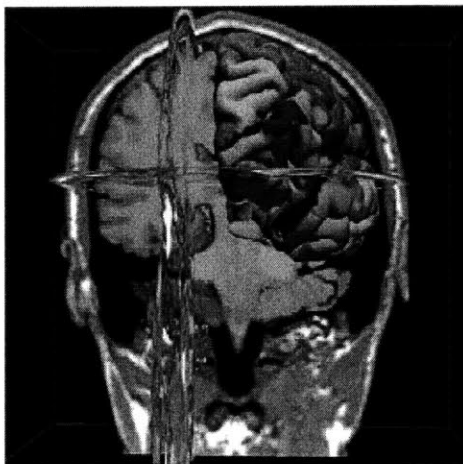


Figure 5-2: Morphometry BIRN frequently uses tools like 3D Slicer for visualizing volumes and surfaces in the human brain

MIT Graduate Student Researchers (MITLS)

The other set of users were graduate students at MIT involved with research in various fields related to life sciences, such as bioinformatics and computational biology.

In contrast to the neuroscience researchers from BIRN, most researchers in this user group do not extensively use direct end user tools for visualization. Their work is very programming intensive in that there is a frequent need to write scripts to query large amounts of data and existing knowledge. A mechanism that simplifies the way queries are specified, saved, edited, reused, and shared is of particular interest to researchers in this group.

Others

The study was also pilot-tested with graduate student researchers who specialized in information management, visualization, user interface design, and user study design. There are also a few other groups of users such as MIT Libraries that have been identified as suitable for future experiments that will study how personalizable end user applications help users with information visualization and management.

5.3.2 Information Collections

In a fully naturalistic setting, participants would use their own information collections during the user study. However, in this more controlled environment, it was necessary to provide test collections for the user to perform the tasks on. In order to make the user

study as naturalistic as possible, we used information collections that closely simulated the real-world environment of the users.

One section of the tasks in the study asked the user to augment their own research data with information from outside resources. To simulate the user's "own research data", we provided a collection of data with the Haystack Evaluation Version. In fact, this corpus of test data that was used in a major part of the user study was real experimental data obtained from BIRN-affiliates, the Internet Brain Volume Database (IBVD).

In an early stage of discussion with BIRN researchers, this corpus of data was provided to the Haystack group to get a better understanding of the data and tools that were used by researchers studying the human brain. At the time this data was given, it contained the entire corpus of brain volume data contained in IBVD. The data was already described in RDF (due to an ongoing initiative to add metadata layers to that database) - this also made it ideal to be used as sample data within the Haystack environment.

The collection *My Data* used in the study contains the data obtained from IBVD. This test data consists of data groups that were used in brain volume experiments corresponding to various publications. A *group* (Figure 5-3) contains information about the number of subjects in that group and potentially useful properties such as mean age, gender breakdown, handedness, etc. In addition, volume and IBVD publication data (which were interconnected through data groups) was also included as sample data.

In addition, to enable users to perform tasks modeled on real-world scenarios, the tools required for frequent tasks such as literature search and graphing was also included in the system. The most useful tools were determined during the initial user-centric design phase. Accordingly, users were given tools to query PubMed and work with results from literature searches. In addition, tools were included in the system for simple graphing and LSID Entrez searches. The design and development of these tools are described in greater detail in Section 4.1.

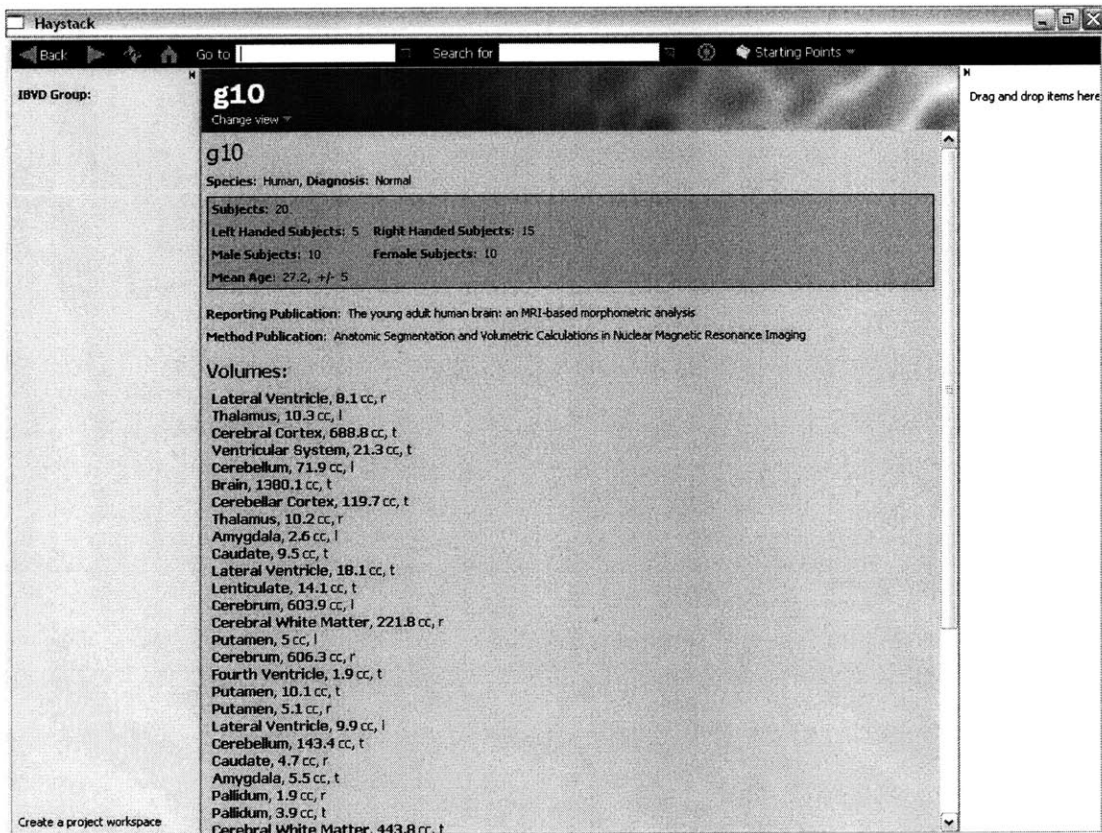


Figure 5-3: A view of a data group in Haystack from the hundreds of data groups obtained from IBVD

5.3.3 Tasks

In order to evaluate the effects of personalization capabilities in the application, we formulated a scenario that encapsulates frequent tasks that target users perform in real life. We had extensive discussions with life science researchers beforehand to determine the type of tasks they do in terms of managing and visualizing their information to make the user study as naturalistic as possible. In addition, during the user study, we asked the user if and how often he was likely to repeat a particular task to gauge whether it was a natural task for the user.

Scenario

The researcher has started working on a new set of brain volume data from the Internet Brain Volume Database. He wants to organize his data in a way that is easiest for his future needs. He also wants to get additional information relevant to this data - he will do so by querying outside databases to connect related information. He will also edit and personalize his information by adding annotations and modifying views so that when he uses this data in the future, it will contain useful information in a way that is most accessible to him.

Task Categories

- *Organizing existing information*

The user was asked to organize his data using collections, bookmarks, sidebars, etc. This included creating new collections, adding information to them, moving and linking the collections in ways the user thinks is useful.

- *Querying databases to get new, relevant information*

The user was asked to query PubMed to search for existing publications relevant to his data. He was asked to explore the results obtained and add annotations as he deemed necessary.

- *Visualizing information in different ways*

The user was asked to look at his data and data collections in different ways using various existing methods for visualization depending on what he found most useful.

- *Modifying default settings for workspaces, views and data*

The user was asked to make simple modifications to existing workspaces and views so that information that he finds most relevant would be displayed more clearly, in the most preferred way. In addition, the user was given the opportunity to augment his old and new data by defining new relationships and adding personal annotations.

5.3.4 Establishing Baselines

To establish a baseline against which the system would be measured, the user was asked to answer a series of questions in the User Feedback Form (Appendix C) that were aimed

at determining how the user currently performed a particular task X, and how the personalizable system under study compared to it. For each of the major tasks, five questions were used to gain information about current methods (Figure 5-4).

<Description of Task X>

a) How often have you needed to perform similar tasks for your work outside of this user study?
 Never Rarely Sometimes Often

b) How do you currently perform such a task? What software, if any, do you use?

c) Describe one way in which Haystack is better for similar tasks than the current method.

d) Describe one drawback of Haystack compared to the current method when used to perform similar tasks.

e) Drawing from your experience from the user study, indicate which method was better for the task.
 Haystack Current method Both the same

f) If one method was better, why or how?

Figure 5-4: Questions in the *User Feedback Form* for each major task, designed to help establish baselines of evaluation

5.3.5 Measures

Since the user study was run on each subject in the presence of a tester, it was possible to gauge standard usability measures such as the effectiveness and efficiency of the system and user satisfaction. In addition, to measure the ease with which each task was completed and the usefulness of the feature itself, the user was asked four questions on a 5-point Likert scale for certain major task components (Figure 5-5).

<Description of Task Y>

a. How easy was it to perform the task in Haystack (1-difficult to 5-very easy)?
 1 2 3 4 5 N/A

b. How useful was having access to this feature (1-not useful to 5-very useful)?
 1 2 3 4 5 N/A

c. How likely are you to do a similar task in the future (1-not likely to 5-very likely)?
 1 2 3 4 5 N/A

d. How often do you need to do a similar task for your work (1-rarely to 5-very often)?
 1 2 3 4 5 N/A

Figure 5-5: Questions in the *User Feedback Form* for certain tasks, requiring response from users on a five-point Likert scale

The user was also encouraged and given ample opportunity to provide feedback about learnability, errors, and general satisfaction with using the system.

It would be possible to assess other measures of usability such as adoption and use, flow, long-term performance, and quality of life only through a long-term evaluation of the next version of the system in a fully naturalistic setting (Section 7.2).

5.4 Overview of an Evaluation Session

After brief initial introductions, the subject was requested to sign an interview consent form and fill out the *Background* section of the User Feedback Form (Appendix C). This

section first asked the user to fill out information such as name (optional), age, gender, educational background, and field of research. Next, the user was asked about the length of experience in his field of research, in using MS Windows and with the World Wide Web. The user was also asked whether he had any prior experience with the Semantic Web or Haystack. These questions helped gain an understanding about the subject's level of experience with related technologies.

Next, the user was presented with an overview of the study. This was either done verbally by the facilitator or by giving the user time to read the Study Overview document (Appendix A).

The user was then given the chance to ask any questions that he might have and presented with the User Study Instructions (Appendix B). This document had two parts: the *Preparation* part gave instructions for users to download and install the Haystack Evaluation Version and stepped the user through a few short tasks to introduce him to the Haystack environment. Part 2 consisted of four sections describing the actual tasks upon which the users were asked to base their evaluation, and an additional section introduced the context for the *Reflection* part of the User Feedback Form.

After completing each section of the User Study Instructions, the user was asked to fill out the corresponding section in the User Feedback Form. This ensured that the user's experience relevant to a particular section was fresh in his memory when he gave

feedback on it and minimized confusion as to where the task boundaries were. As discussed in previous sections the format of the questions after each task took the form shown in Figure 5-4 and Figure 5-5. These questions helped establish baselines of evaluation, quantify user feedback, and gave the user an opportunity to give open-ended feedback. The final section in the feedback form gave the user the opportunity to reflect on different parts of the application and provide any additional suggestions or comments.

After the user had completed the tasks and filled out the feedback form, the observer asked a few questions about anything that surprised the subject about Haystack and any particular problems he had encountered during the study. After further discussion, the user study session ended.

Chapter 6 Experimental Results

An equal number of users from the two main groups that were introduced in the previous chapter participated in the study. This chapter discusses the reactions and comments given by users during these sessions and analyzes the data that was obtained.

Users were generally quite positive about the flexibility afforded by the personalizable elements in the system. All the life science researchers from the different groups acknowledged the lack of an application, such as the one that was being evaluated, to manage and visualize information in ways that they personally feel will facilitate their work the most.

In addition, all the life science researchers who participated in the user study expressed interest in continuing their collaboration with the Haystack group to follow the progress of the system in the future. They felt that after seeing this version of the system, they would be able to give very useful feedback in future evaluation sessions. The subjects in the study were also very interested in learning more about the different variants of Haystack and other applications related to the Haystack group, which also put an emphasis on information visualization and management.

6.1 General Observations

The user study itself was pilot-tested with members from the Haystack group who were familiar with the system, in order to eliminate errors and identify potential problem areas. Nevertheless, there were still a few unforeseen observations during the course of the actual user study when it was run on the intended subjects.

Even though step-by-step, detailed instructions were given in the User Instruction Documents (Appendix B), some users sometimes found that certain things were still not obvious because they had not seen the system before. On the other hand, other users commented that it was not necessary to give so many instructions. However, on the whole, the opinion was that the instructions were useful and served its purpose well. The problems that users ran into are discussed under Usability Issues (Section 6.3.3).

A few users skipped the last tasks as they were tired towards the end of the study. Although care was taken to keep the sessions as short as possible, sometimes outside influences such as the time of the day, the day of the week, the mood of the user, and how much time the user had beyond the scheduled time greatly affected the outcome of a session. For instance, one user took over two hours to complete the assigned tasks, pausing frequently to ask questions from the observer and to write detailed comments on the feedback form. On the other hand, one user had a meeting right after the scheduled user study and this made him rush through the last parts of the user study.

Some potential subjects who had agreed to participate in the study, failed to follow-up when asked to specify a time for the session. In one case, when one user took double the estimated time to fully experience the system, the observer was obliged to cancel the session that was scheduled after it, since it would have been counterproductive to cut short any session.

Subjects who generally use Windows or Macintosh operating systems found it easier to get used to the Haystack environment. They were already somewhat used to the idea of context menus, menu bars, drag-and-drop manipulation, etc. Subjects who usually work with command line interfaces had a slightly more difficult time being introduced to the system as they were less accustomed to mouse-intensive tasks and certain conventions used with components like folders (identification by yellow folder icon), drag-and-drop (how to move items from one place to another), and context menus (menu items determined by relevance).

It was more difficult to run the user study with experienced researchers who were relatively set in their ways. Such users were more inclined to question the motivation behind a task assignment in the user study, even though it was counterproductive to make them fully aware of what exactly was being observed and evaluated, beyond having a general idea about the high-level focus of the user study.

Figures 6-1 to 6-4 show screenshots of collections and data created and used by subjects, collected at the end of certain evaluation sessions.

6.2 Analysis

The number of subjects used in the study was not sufficient to test hypotheses with statistical significance. However, for the purposes of a formative evaluation, it was possible to use the data obtained to do a satisfactory qualitative analysis of the system.

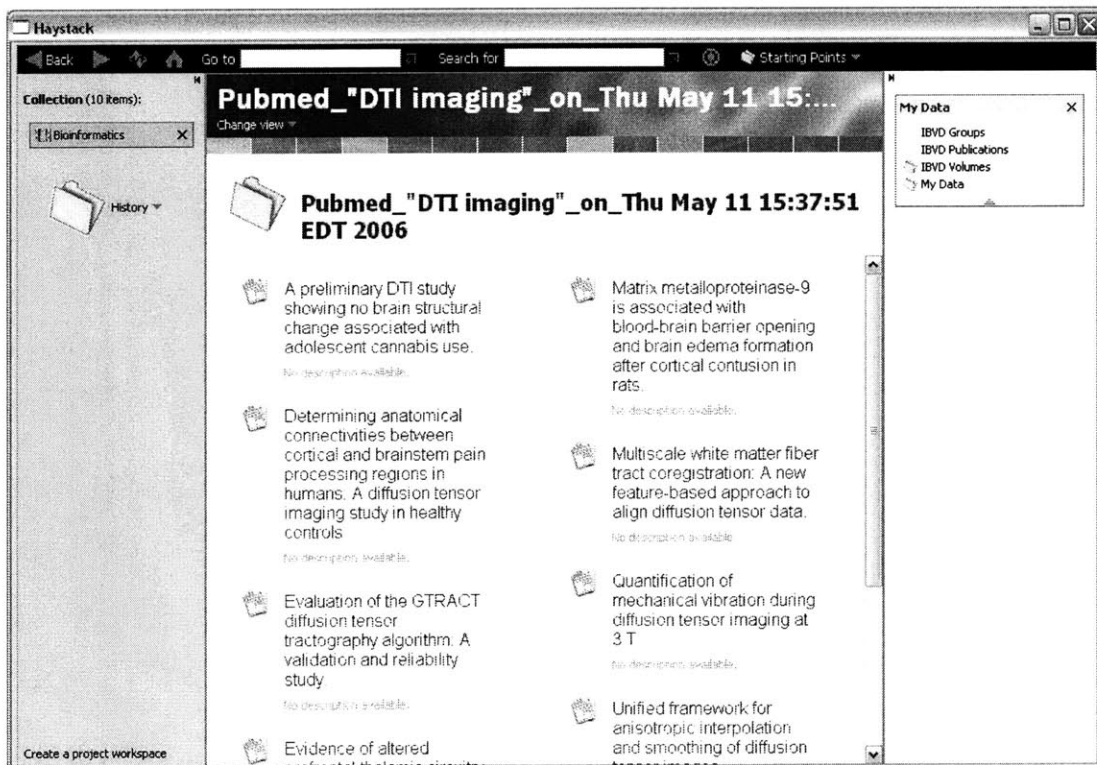


Figure 6-1: A screenshot from a user session: PubMed results obtained for the search “DTI imaging”

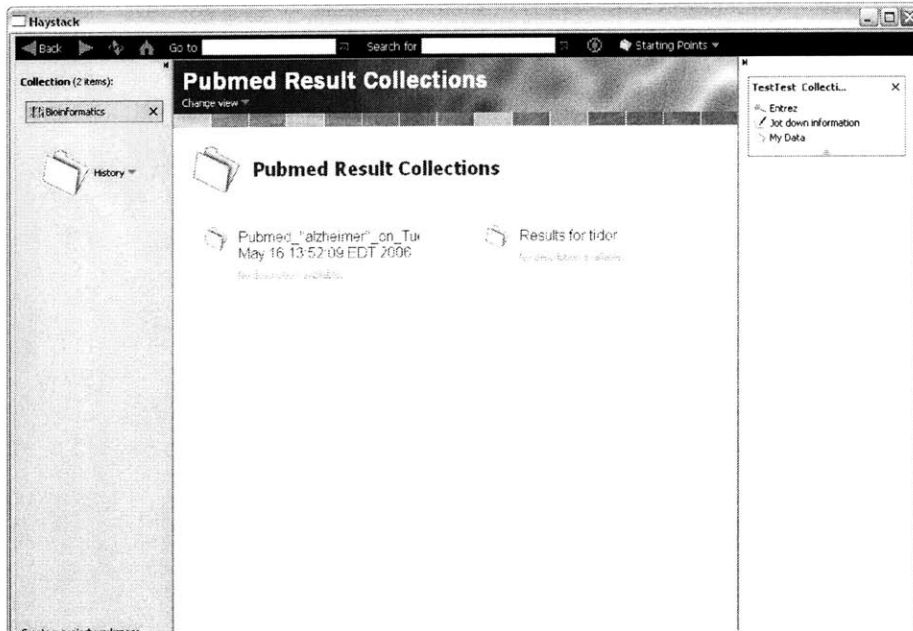


Figure 6-2: A screenshot from a user session: collection containing records of all past PubMed queries and results

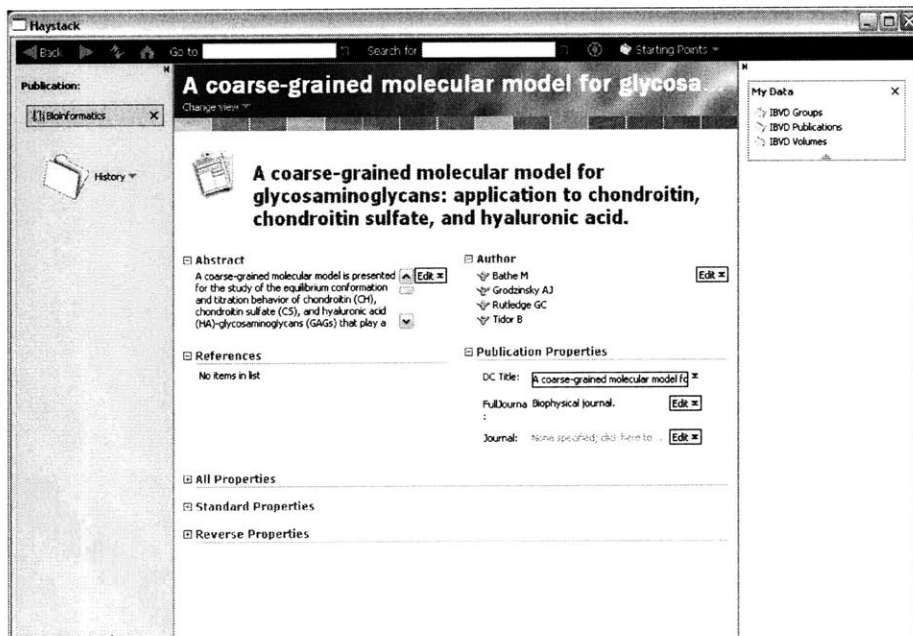


Figure 6-3: A screenshot from a user session: the personalized version of the preferred view of a publication

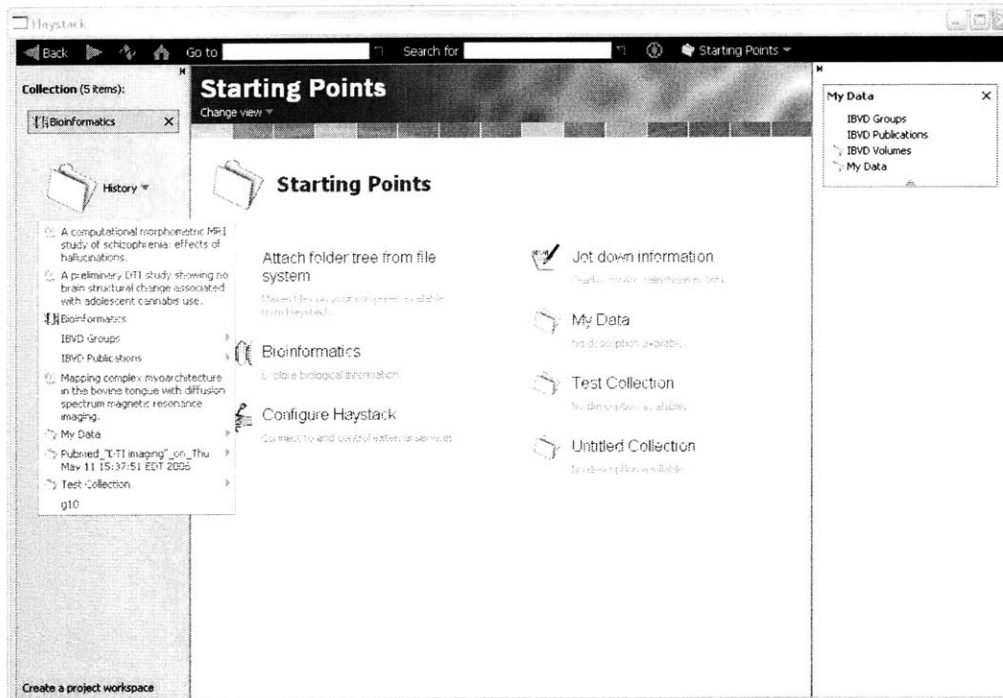


Figure 6-4: A screenshot from a user session: browsing history

6.2.1 Main Tasks

In this section we report user responses to the main tasks that they performed as part of the user study. We present four main tasks and discuss the users' written responses and feedback given to the observer. Tables 6-1 to 6-4 show the written responses of a group of users (filled out in the feedback form after completing each task).

Key for Tables 6-1 to 6-4, columns a through f:

- a) How often have you needed to perform similar tasks for your work outside of this user study
 - b) How do you currently perform such a task? What software, if any, do you use?
 - c) One way in which Haystack is better for similar tasks than the current method.
 - d) One drawback of Haystack compared to the current method for performing similar tasks.
 - e) Which method was better for the task?
 - f) If one method was better, why or how?
-

Managing your data so that you can organize your thoughts and find important information at a later date

In this task, users organized data by manipulating collections and links. The current method for doing similar tasks was different for different users (Table 6-1): Microsoft Windows, Internet Explorer, text editors like EMacs, good directory organization, etc. were cited by users as methods they currently use to organize their data according to personal need and preference.

Table 6-1: Responses for “Annotating and managing your data so that you can organize your thoughts and find important information at a later date”

Frequency	Current Method	Haystack Advantage	Haystack Disadvantage	Which is better?	Why?
Sometimes	Lab notes, notebooks, good directory organization	Haystack has multiple links to the same data	Haystack too new to answer	Both the same	
Often	IE, Windows		Not sure how to add things outside of the app	Both the same	
Often	I don't use one piece of software for this purpose. Instead, I make multiple copies of data/ folders/ figures for multiple projects. Or use scripts in UNIX/ LINUX linking files.		Not enough experience to say. However, I don't find the interface visually intuitive.	Not enough experience to evaluate	
Sometimes	Text files & custom scripts, EMacs	Comprehensive & intuitive	Less flexible	Both the same	
Never	Currently do not use any special software. Organize my folders according to task categories	More flexibility in adding different types of objects to folders	Repeated instances of collections on different panes are somewhat confusing at first	Haystack	See (c)

The users didn't see the annotation capabilities in Haystack until they completed the next task, so they were responding to how information is managed in Haystack (with collections, links, etc.) when answering this question and comparing it to how they do similar tasks with current methods.

All test users found the Haystack system to be at least as good as current methods but noted during discussions that the features afforded in Haystack were more powerful and flexible. Subjects thought that with more experience with the system, they would probably prefer it to the old methods.

Querying databases related to your research; annotating and organizing results from your search in a way that will be easy to retrieve in the future

In this task, users queried PubMed, viewed the results, and edited the results collection as preferred. This was the task that users felt was most useful and easy to perform using Haystack. Users especially liked the control afforded by the application to organize and annotate new information obtained from outside databases. Users found the ability to save and organize relevant results and also add their own notes to the data to be a very useful feature. All the users who responded found Haystack to be the better method for this task (Table 6-2).

Users noted that current methods include using a web browser and jotting down, bookmarking, or writing down useful information. In addition, for the particular task of

literature reference management, some users mentioned that they currently use EndNote, a specialized commercial software tool for publishing and managing bibliographies. EndNote (Section 2.4.2) is highly specialized for the task of managing bibliographies and users noted that Haystack has broader functionality in terms of the types of information that can be managed within the application.

Table 6-2: Responses for “Querying databases related to your research; annotating and organizing results from your search in a way that will be easy to retrieve in the future”

Frequency	Current Method	Haystack Advantage	Haystack Disadvantage	Which is better?	Why?
Often	Web-based use of PubMed, print results or store them in EndNote	Results are saved w/ study w/out needing to specify a new filename of location	Maybe harder to manage <u>hundreds</u> of references...	Haystack	Better way to save results
Often	Web browser	Can edit results in the app	Outside of normal browser		
Sometimes				Need more experience	
Often	PubMed on the web & simply bookmark links	Haystack makes the folder of results more readily available		Haystack	Haystack provides more functionality & is easier to use & right to the point
Sometimes	I do not use software. I generally use bookmarks or jot down references.	It provides better organization of many query features.	There is maybe too much displayed.	Haystack	See (c)

Visualizing your information in multiple ways, depending on the task at hand

In this task, users viewed resources of different types in different views and modified the publication view by adding a new “pane” to make author information more visible. For most users, the idea of changing the layout of part of an application was a new concept to consider. A few users noted that this functionality could potentially be very useful after they had more experience using the system (Table 6-3).

Table 6-3: Responses for “Visualizing your information in multiple ways, depending on the task at hand”

Frequency	Current Method	Haystack Advantage	Haystack Disadvantage	Which is better?	Why?
Sometimes	Reorganize it. - none-	Multiple links to the same data/ information	Not sure yet	Both the same	
Often	database tools, web search interface (PubMed)				
Sometimes	Depends on data. No one set approach.	Need more experience		N/A	
Sometimes	None	It's flexible & structured at the same time	N/A	Haystack	Haystack is better because before, I would simply filter in my head
Rarely	I do not use software, and I currently don't perform this task.	Allows this functionality; applies it broadly over all search panels.	I wasn't aware that the changes would be applied broadly. Not sure if an undo function exists.	Haystack	See (c)

In subsequent discussions, one user described how it is necessary in his research to visualize the brain at different layers of abstraction and different perspectives (such as physical structure, function, interaction, etc.). He currently did this using different layers of tracing paper that could be removed or added, depending on the layer of abstraction and amount of detail that was necessary. He was intrigued by the idea of multiple views and perspectives in Haystack to enable him to visualize information in ways he found “visually intuitive”.

Users stated that they currently do not have a set method to perform this task and applications that provide close to similar functionality do not present the same flexibility. Overall, users found the Haystack system to be the better method.

Defining and exploring relationships between different data sets

The responses regarding the features in Haystack that enable users to create and visualize connections between data sets were very positive (Table 6-4). Users noted that currently there was no effective way to add or edit information to personal copies of outside data and connect them to existing data in a way that would be usable at a later date. The particular task that they did in Haystack involved viewing the reference information for an IBVD publication (which simulated the user’s own data) and using an operation to retrieve the full abstract and standard reference information from PubMed so that the two sets of information was linked.

They admitted that this was a task that they found very useful and expected that if such features to personalize the data model were available in an application they would perform similar tasks often.

Table 6-4: Responses for “Defining and exploring relationships between different data sets”

Frequency	Current Method	Haystack Advantage	Haystack Disadvantage	Which is better?	Why?
Sometimes	Download paper from full-text (online) source, or view it via PubMed or EndNote	Seems like a complete reference to the article	Sometimes need full-text article to browse Tables, Figures, and Discussion section of an article, or browse reference section	Current method	I get more information from the actual publication
[skipped]					
Sometimes	For references? EndNote. Wikipage for selected groups	Very efficient, maintains custom format.	Not enough experience to know.	Need more experience	
Rarely	Just Google	Automatic	None	Haystack	Haystack is better because it does the work for me (automatically)
Sometimes	I do not use software	Provides a simpler and clear way to connect data sets.	I don't see any.	Haystack	Previously, I didn't have a good way to pull up different data sets easily by clicking a link.

6.2.2 Numerical Data

Using questions that required answers on a Likert scale made it possible to quantify user feedback. The responses from users for this section are shown in Table D-1 in Appendix D. The average rating given by all users on a 5-point scale for each task is shown in Table 6-5.

The standard deviation of the sample means (average ratings) is found by calculating the standard error of this data.

$$\text{Standard error} = \frac{\text{Standard deviation}}{\text{Square root}(\text{Total number of data points})}$$

The standard error (ϵ) of each measurement is shown as $X \pm \epsilon$. The numbers in boldface indicate a measurement where one user had skipped answering the question.

Table 6-5: Average ratings and their standard error for ease of use, usefulness of feature, likelihood of repeating similar tasks, and the estimated frequency of repetition

Task	Ease of use	Usefulness	Likely to repeat	Estimated Frequency
Viewing all application windows necessary to complete a task on the screen at once (11g)	4.5±0.22	3.6±0.22	4.2±0.34	3.6±0.47
Querying a database and annotating and sorting the results (12g)	4±0.41	4.75± 0.22	4.8±0.18	4.2±0.34
Visualizing data of the same type in different views (13g)	4± 0.34	4.2±0.18	4.2±0.18	3.4±0.37

Creating a new way of visualizing data of a certain type (13h)	3.8±0.34	4±0.29	3±0.29	3±0.29
Making persistent connections between different data sets for future use (14g)	4.25±0.43	4.5±0.26	4.25±0.22	3.5±0.26

When analyzing responses to a Likert scale question, a rating of 3 or higher is considered to indicate that the user agrees with the assertion in question.

On average, subjects agreed that the ease of using the application for the selected tasks is high. Subjects thought that it is very likely that they will repeat the task that they just performed. However, the estimated frequency with which subjects think they will perform these tasks is somewhat in the mid-range.

Depending on the task, this phenomenon can be interpreted in different ways. For instance, the average rating for the estimated frequency of repeating the task “Creating a new way of visualizing data of a certain type” is 3 - it can be argued that this is because once a view is personalized, this is not a task you would expect to repeat often. However, you can expect to perform the function from time to time. Another point to note is that the standard error of the average ratings for estimated frequency of repetition is high compared to the standard error for ease of use, usefulness, and likelihood of repetition for corresponding tasks. This may be an indication that the two user groups that were

involved in the study had varying responses for this question. This is discussed in further detail at the end of this section when *Comparing Groups*.

The task that got the highest rating was “Querying a database and annotating and sorting the results”. Subjects from all domains recognized this as a familiar task that is currently performed with tools that have limitations. All the users in the study had used PubMed or other Entrez databases for their work and had been compelled to save results or bookmark pages manually, without being able to view any connections between different results sets. In Haystack, the results are saved as it is received from the query and connections between datasets are automatically defined because of the way unique names are handled in that environment.

Several subjects were very surprised by the ease with which information from outside databases could be brought into the application to be viewed in customized ways and organized as the user saw fit. For instance, when they received results back from a PubMed query, the titles were displayed in a new collection as shown in Figure 6-1. They could view each publication in greater depth and bring up interesting properties to create new panes in the view.

In addition, the idea of adding persistent personal annotations was found to be a task that most users always wanted to perform. In Haystack, they could add or edit the results that

are saved from database queries so that they could be found in the future without having to be dependent on text files and written text to keep track of personal notes.

It is important to recognize here that we are analyzing the average over *all* subjects, which includes users of two distinct groups. It is interesting to consider the average responses of these two user sets separately.

Group 1 – BIRN Researchers

BIRN researchers thought that the likelihood of them repeating the tasks that they performed is high (Figure 6-5). This indicates that the formulated tasks were appropriate for the target domain. The error bars on the chart indicate the standard error of the sample mean.

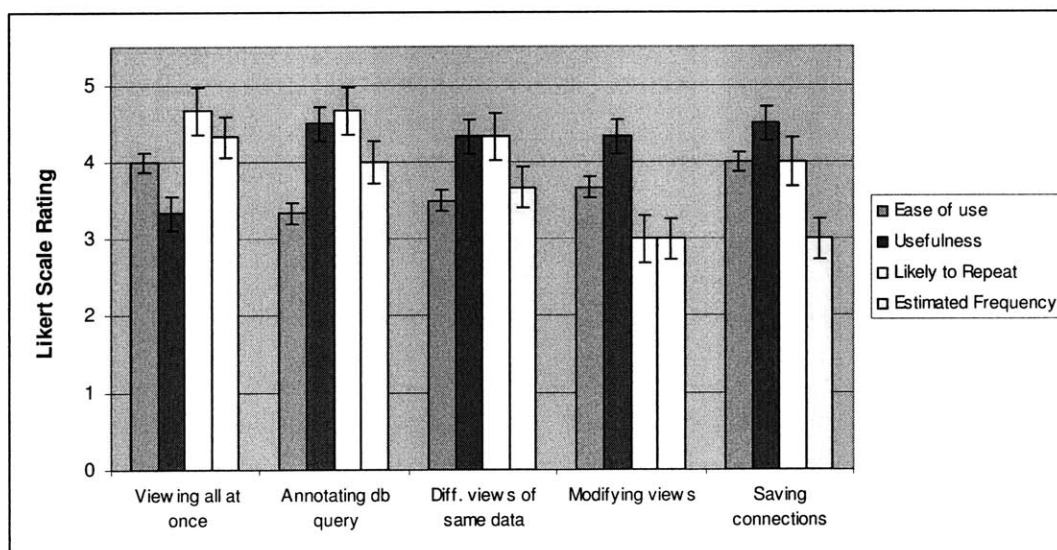


Figure 6-5: The average rating given by BIRN researchers

Group 2 – MIT Graduate Student Researchers

MIT graduate students found the tasks comparatively easy and the features useful (Figure 6-6). However, they did not expect to repeat the tasks at a high frequency in the future. This again makes sense because the tasks were formulated after observing life science researchers and determining what type of tasks they frequently need to do. The error bars on the chart indicate the standard error of the sample mean.

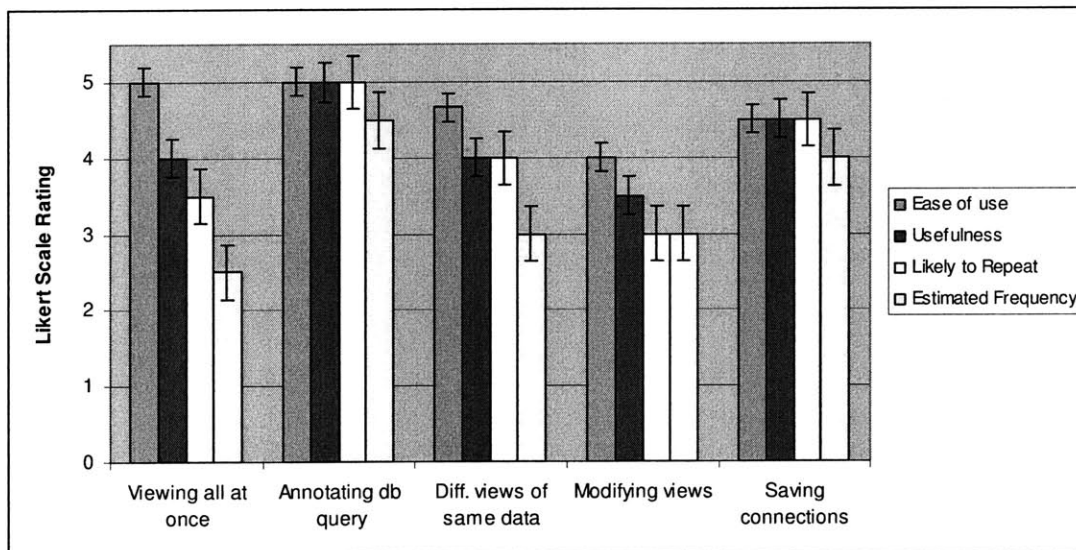


Figure 6-6: The average ratings given by MIT graduate student researchers

Comparing Groups

Figures 6-7 to 6-11 directly compare the responses from BIRN users and MIT users for each task. The general trend shown here is that MIT users found the tasks easier than

BIRN users. On the other hand, BIRN users thought it was more likely that they repeat the tasks and expected to repeat these tasks at a higher frequency. Both groups agreed equally that the features were useful. The error bars on the chart indicate the standard error of the sample mean.

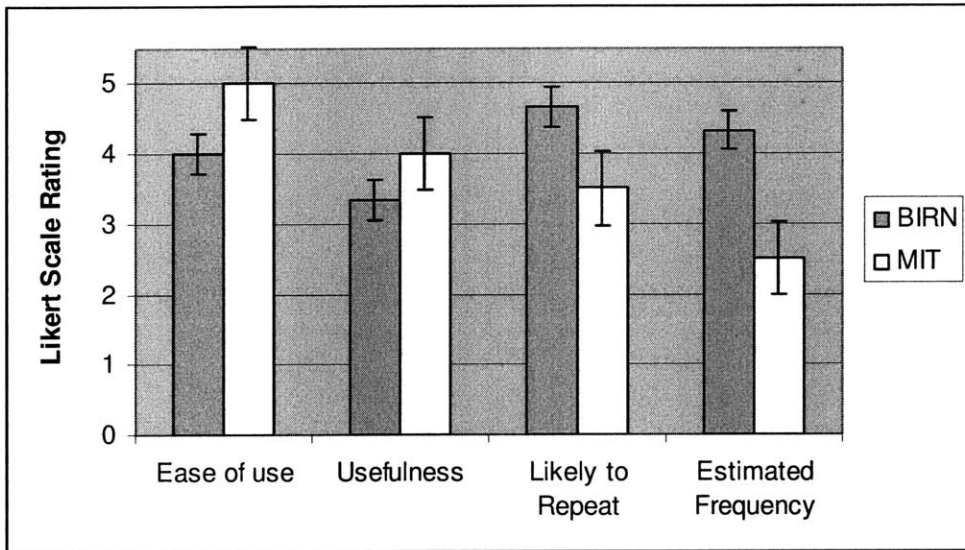


Figure 6-7: Comparing group average rating for the task, “Viewing all application windows necessary to complete a task on the screen at once (11g)”

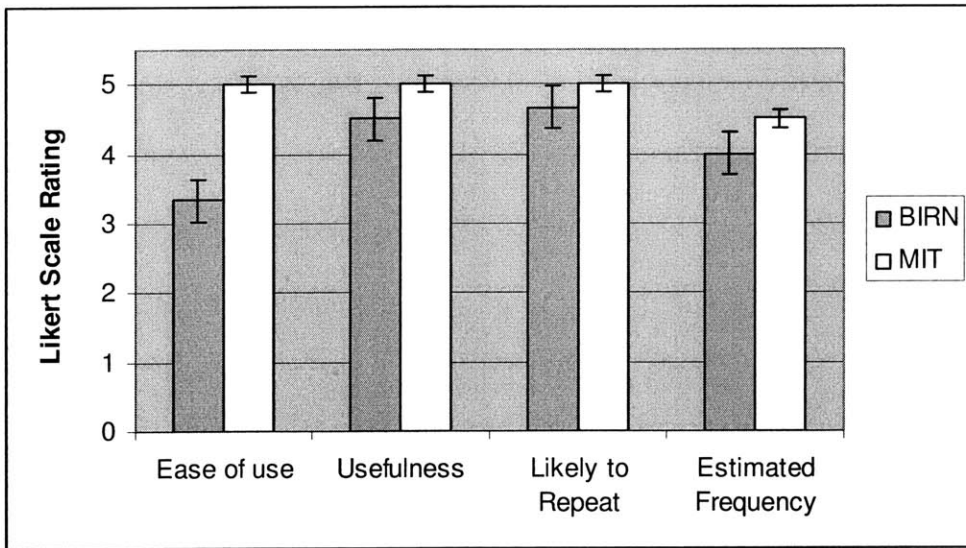


Figure 6-8: Comparing group average rating for the task, “Querying a database and annotating and sorting the results (12g)”

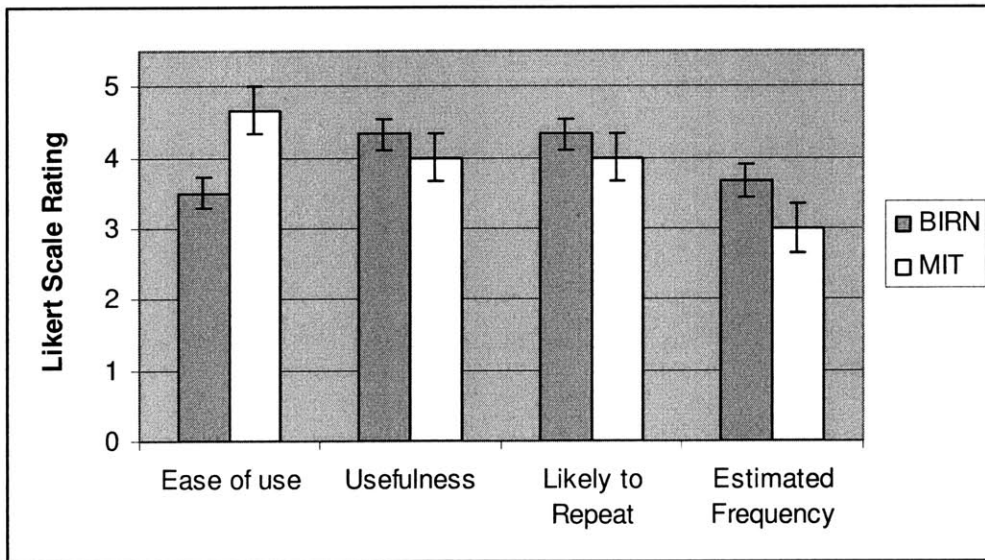


Figure 6-9: Comparing group average rating for the task, “Visualizing data of the same type in different views (13g)”

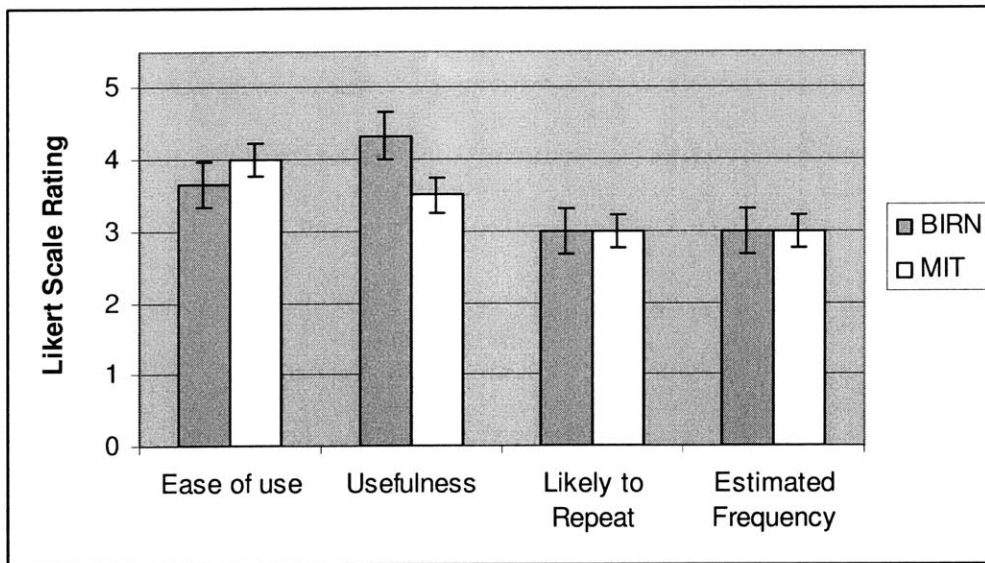


Figure 6-10: Comparing group average rating for the task, “Creating a new way of visualizing data of a certain type (13h)”

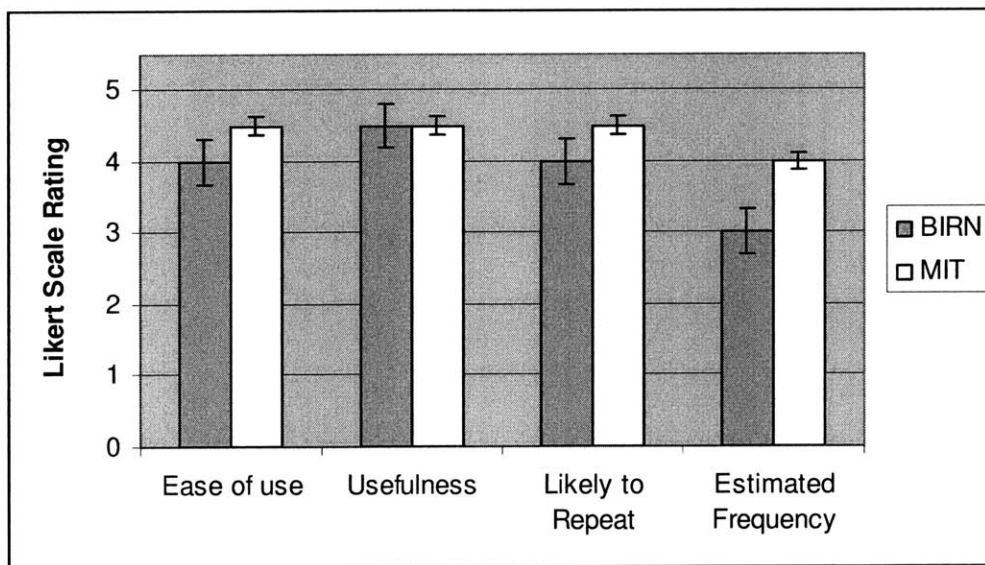


Figure 6-11: Comparing group average rating for the task, “Making persistent connections between different data sets for future use (14g)”

6.3 Design Implications

Certain questions in the User Feedback Form were designed to determine the user's thoughts about what works and what doesn't in the current version of the Haystack system. The responses to these questions gave suggestions and comments that could potentially be incorporated in to future versions of the system.

Due to the fact that an observer was present during the user study sessions, it was also possible to interview the user right after the session and have a continuing dialogue with users as they generally had more comments or asked for more information. The interview and subsequent discussions yielded helpful insight into the user's experience with the system and their opinion of various aspects of it. This was especially useful when a user hadn't been very descriptive in his written responses.

In this section, we discuss the users' viewpoints on the strengths and weaknesses of the current system (including usability problems encountered by users during the evaluation session), and consider the design implications of these findings.

6.3.1 Strengths

Users cited the flexibility given to the user as a factor that made the personalizable application better than current applications that they use. Users commented on how the system was “automatic”, “flexible, yet structured” and was “intuitive” to use.

Some MIT researchers who were part of the user study were very interested with the query mechanism in Haystack. The evaluation sessions demonstrated to them how Haystack can help users to query outside databases and view the results in useful ways. The users thought that the features which enabled editing, annotating and saving these results for future use had a lot of potential.

6.3.2 Limitations

Most users thought that they haven't had enough experience to encounter limitations of the system. One user mentioned that he was unsure how to bring outside data into Haystack and commented that it was not possible to drag and drop into and out of the application. A user also felt that it would be very useful if Haystack actually ran inside of a browser and cited the Semantic Web Application in Neuromedicine (SWAN, Section 2.4.4) as an example of such an application.

6.3.3 Usability Issues

- In the detailed instructions that were given, although screenshots were shown, the exact position of buttons and links were not specified. This was done intentionally to test how well users could find things on the screen. Until users got used to the environment, it was rather difficult for them to find certain things. The change view button and the forward button in particular had low visibility.
- In general, users found the preview pane distracting or confusing, and tried to ignore it when performing their tasks. One user commented that the preview pane was generally not useful but the user's main focus is sometimes inadvertently drawn to it. They were not able to close it on their own.

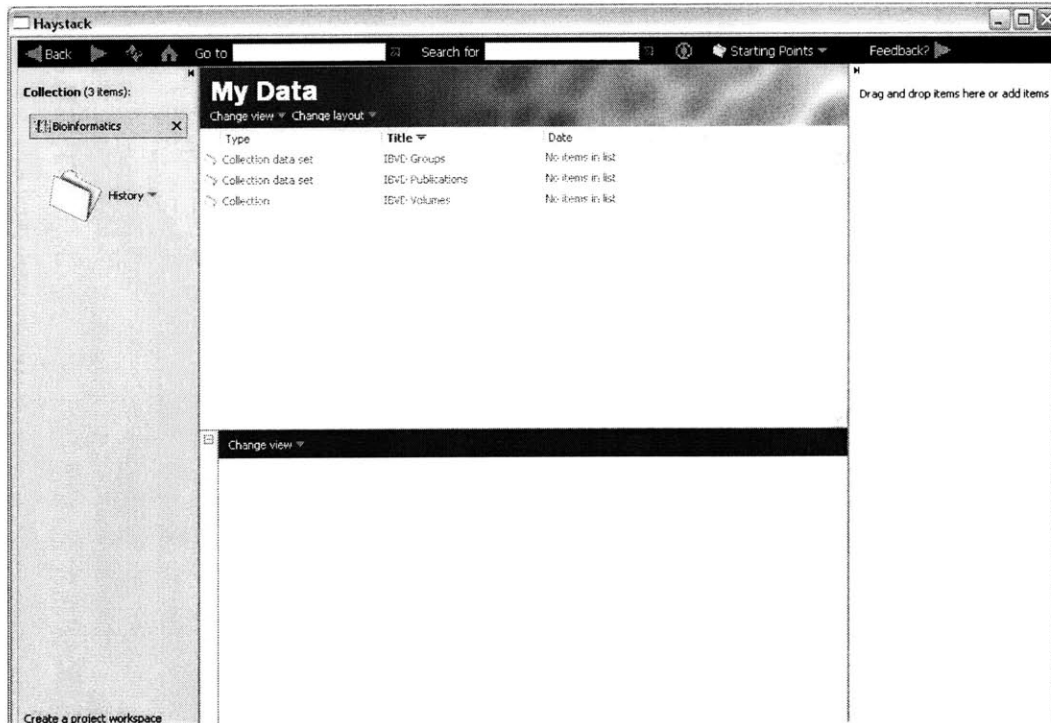


Figure 6-12: Default view with preview pane

- One reason that the change view and back/forward buttons have lower visibility could be due to the dark background of those menu bars. One user commented that the use of colors were not “intuitive” and did not show enough contrast between different parts of the system.
- Some users commented that the interface for performing operations was cumbersome. The sequence of steps was considered unnecessarily long: (right click on operation and find “Perform operation” in the right submenu, move to the right pane, click *Edit* for each required input, click the text box to get keyboard focus, enter each input, press enter after each input, click *Edit* again after each input, and click *OK* to perform).

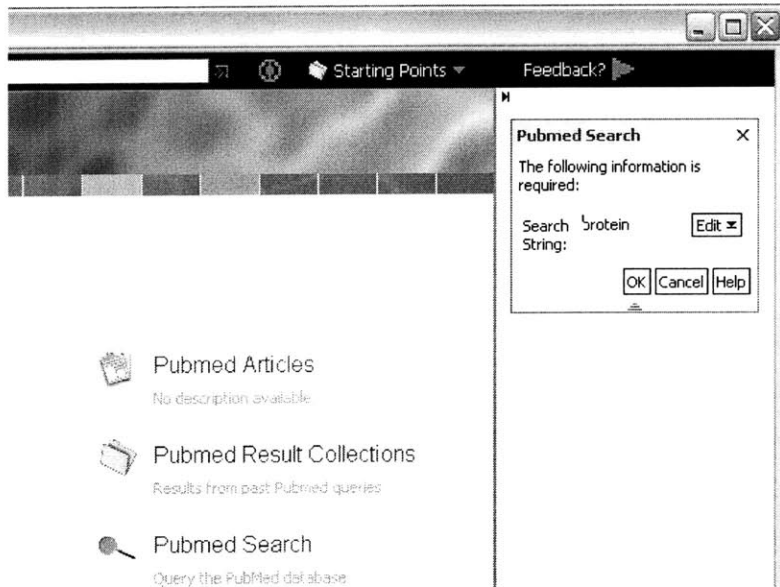


Figure 6-13: Slide to enter required inputs for an operation in the Haystack right pane

- The order of type and name columns in the Table view is the opposite of what is generally the standard (name, then type). Some users questioned the reason for this difference. It is an unnecessary attribute and can easily be changed in future versions of the system.
- Items within collections did not appear to have a useful order to the users (in Browse view). This made it more difficult for them to find items. It was also not possible to change the order according to user preference. In addition, the position of items sometimes changed within the collection and this diminished user recall.
- Context menus were confusing to even the most advanced computer users (Figure 6-14). The menu for even the same resource changed according to small changes in

context (that were not visible or relevant to the user). This meant that every time a user performed a certain operation, the necessary menu item had to be found each time by going through submenus. This led to low consistency and again deterred user recall.

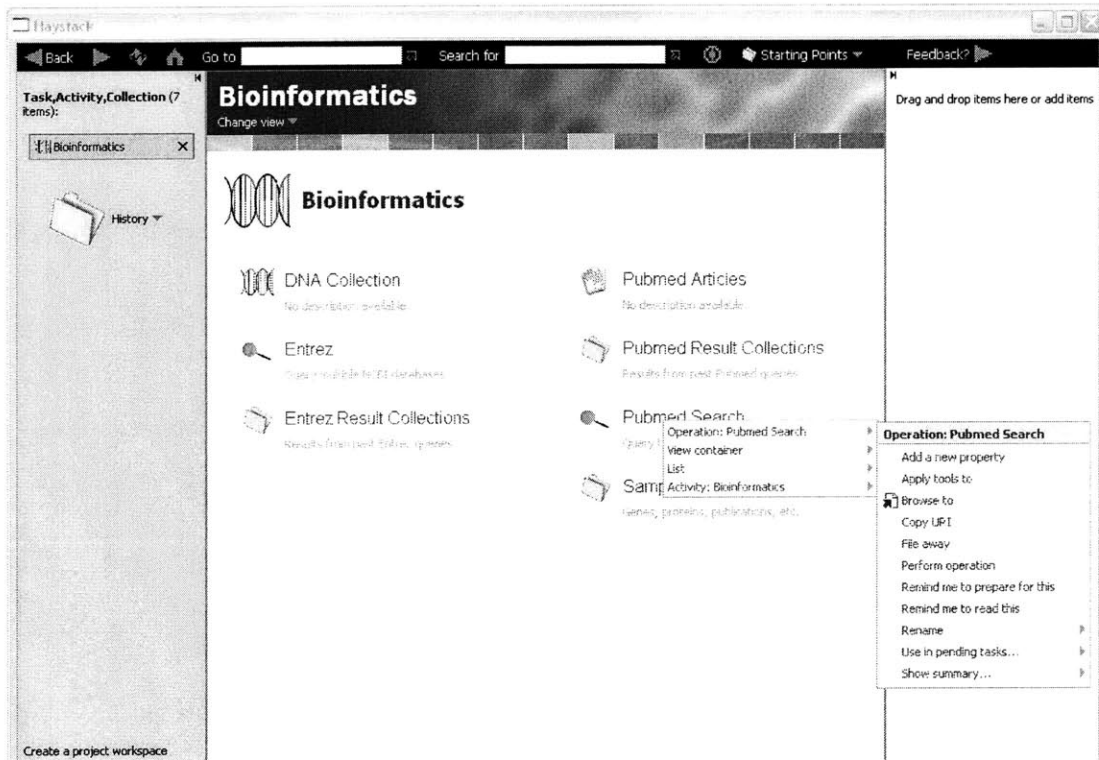


Figure 6-14: A context menu, showing submenus for different levels of abstraction

- Some users did not like the names of the menus and menu items. Titles like “View container...” did not convey useful information to the user because it is simply exposing unnecessary developer terminology. It was also noted that the number of menu options could be cut down to major or essential ones.
- System messages were easy to miss because they were presented like anything else on the right pane and appeared to be just another resource being displayed.

- At least one user commented on the fact that errors he made during the user study were not reversible. The lack of an adequate undo mechanism made it harder for users to recover from errors.

Chapter 7 Conclusion

This chapter presents the conclusions of our research: the first section discusses its contributions while the second section identifies directions for future work by suggesting potential avenues for continued development and evaluation.

7.1 Contributions

During the course of this project it was possible to recognize and define a specific problem that researchers face in the form of inflexible applications that don't recognize or allow for the needs of individuals. After identifying the problem space, we were able to look at ways in which this issue could be resolved.

The in-depth study of relevant background work and the exploration of related domains were useful in determining where past attempts have failed and where they have succeeded when creating personalizable applications to aid researchers in their work. In addition, we were able to conclude from this exploration that semantic technologies were well suited to solve problems in bioinformatics.

We were able to incorporate what was learnt through our background study into an application built on Haystack that contained the tools necessary to demonstrate the

benefits of personalization and semantic technologies to aid life science researchers with their work.

Extensive discussions with bioinformaticians and observation of life science researchers helped to formulate realistic scenarios and tasks that could be used to evaluate the system. The tasks were designed in such a way as to compare the application to current methods used, and determine how the personalization capabilities enhanced the user experience and increased their effectiveness.

The user study introduced life science researchers in the community to new concepts regarding the way their data and knowledge can be organized and viewed. The current version of system was well-received and those who evaluated it recognized its potential.

By analyzing the results obtained from the user study, We were able to determine that personalization capabilities were indeed useful to users. Working with different parts of the system that was developed showed the user the potential to improve information management and visualization through the use of appropriately designed personalizable applications.

7.2 Future Work

This section presents possible directions for future work that build on the findings of this thesis.

7.2.1 Developing the Query Builder

Haystack currently has the backend capability to support powerful queries. However, this functionality is not fully exposed to the user in a useful way. From our experience with talking to researchers about the Haystack environment, we think it would be very helpful to give the user access to these querying mechanisms to improve the ways in which they can manage and analyze their data within our system. This can be implemented through a simple interface that resembles existing UIs that support filtering to enhance user recognition and learnability.

Figure 7-1 shows the interface in iTunes that enables users to create “smart” playlists that are dynamically updated according to the filtering criteria that is specified. This is similar to dynamic collections in Haystack, where currently the filtering criteria can only be specified by writing Adenine code. The interface shown is simple but handles rather complex logical statements. Such an interface in Haystack would make it easier for users to use the querying functionality in Haystack.

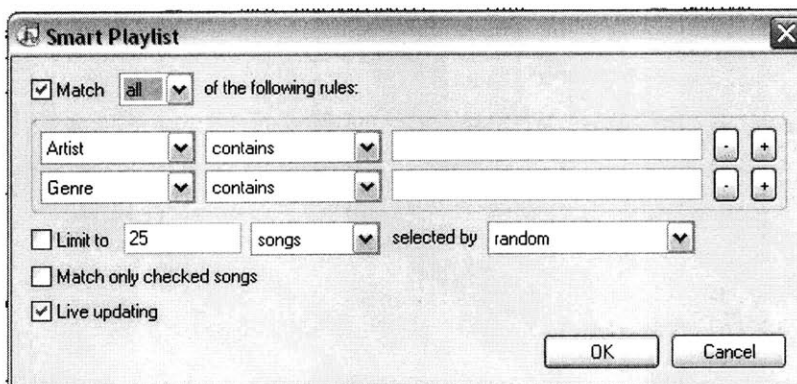


Figure 7-1: Interface to create a “Smart Playlist” in Apple’s iTunes

Currently there are two resource types in Haystack that can be used to keep track of query results: collection data sets (dynamic collections) and “channels”. Figure 7-2 shows the specification of a collection data set in Adenine.

```
add (:allJournals
    rdf:type          data:CollectionDataSet ;
    dc:title          "All Journals with PubMed publications";
    data:dataSource ${
        rdf:type          data:RDFQuerySetSource ;
        data:existentials @ ( ?x ?y ) ;
        data:targetExistential ?y ;
        data:statement ${
            data:subject ?x ;
            data:predicate rdf:type;
            data:object    ncbi:types:pubmed
        };
        data:statement ${
            data:subject ?x ;
            data:predicate ncbi:predicates:journal;
            data:object    ?y
        }
    }
)
```

Figure 7-2: Adenine code that specifies a collection data set or dynamic collection in Haystack

This particular dynamic collection keeps track of all the journals that every PubMed article in the Haystack RDF Store has been published in. I.e., it filters for the object related to resources of type `ncbi:types:pubmed` with the predicate `ncbi:predicates:journal`. Approximately four to six logical statements can be joined together to specify the filter corresponding to a dynamic collection. This is sufficient functionality to package into a powerful query interface for research data.

The channel ontology specification and related operations are mainly used at present in specifying workspace content. The interfaces for their use are also not very usable. One

thing that could be changed is the terminology used in this specification so that it will match what users are generally used to (such as renaming “channels” to something like smart filter or persistent query).

7.2.2 Changes to Workspace Layout

The intended aim of the workspace manager is to enable the user to view as much relevant information on one screen as possible to aid in performing a particular task. However, in the current implementation, the right and left panes are generally not used to convey any information that is useful in the context of that particular workspace. Future implementations should consider using the layout of Haystack in a more efficient manner to arrange as much information as possible according to the requirements of the user.

7.2.3 Extended User Evaluation

The next iteration of the system can be built after considering the feedback obtained from our user study. In contrast to the evaluation done in this thesis, it would be most useful for the next stage to be a long-term evaluation of the system with similar users in a fully naturalistic setting. This can be done after incorporating the revisions based on the analyzed results and user suggestions. Such a study should demonstrate the practical capabilities of the system and it would enable the assessment of usability measures such as adoption and use, flow, long-term performance, and impact on quality of life.

Appendix A – User Study Overview

Thank you for agreeing to participate in the Haystack User Study. In the course of this evaluation, you will be asked to use and comment on a new version of Haystack. Instructions for downloading and installing the software can be found with the *User Study Instructions* document that has been provided to you.

What is Haystack?

Haystack is a tool designed to let individuals manage all their information in ways that make the most sense to them. By removing arbitrary barriers created by applications that handle only certain information "types" and record only a fixed set of relationships defined by the developer, we aim to let users define whichever arrangements of, connections between, and views of information they find most effective. Such personalization of information management will dramatically improve everyone's ability to find what they need when they need it.

Requirements (recommended)

- Windows XP
- 500 MHz Intel Pentium III processor or greater
- 250 MB available disk space
- At least 512MB of RAM
- High-speed internet connection

Estimated time requirement: 1 hour

Resources

You should have a copy of the following documents:

- User Study Instructions
- User Feedback Form

In addition, you can use the feedback mechanism in Haystack to send error reports or get help directly (by emailing hs-birn-study@mit.edu) if necessary.

More detailed documentation on Haystack can be found online:

<http://simile.mit.edu/hayloft/>
<http://haystack.csail.mit.edu/documentation.html>

Next Steps

Please follow the guidelines provided in *User Study Instructions* to complete the user study. After completing the tasks given to you, please fill out the *User Feedback Form* and return to Sumudu W. Watugala no later than May 12th 2006.

Appendix B – User Study Instructions

This user study has 2 parts:

- The 1st part involves downloading and installing the Haystack Evaluation Version (if it is not already installed in the computer that you are using) and completing a few tasks to familiarize yourself with the Haystack environment.
- The 2nd part steps you through the main tasks that will help you think about how the functionality in Haystack can help in the research process by improving information management and visualization. After each major task in this section, you will be asked to fill the corresponding section in the User Feedback Form.

Your feedback is very important to us. Please take notes as you go along if you think this will help you remember your impressions about the Haystack experience. You can submit your comments and suggestions with the User Feedback Form.

Skip to page 3 if you are working on a computer that has Haystack Evaluation Version already installed.

Part 1 - Preparation

The first 2 tasks involve downloading and installing Haystack and need to be completed only once. The rest of the tasks in Part 1 will help you familiarize yourself with the Haystack environment and should take around 15 minutes to complete.

1. Download Haystack Evaluation Version

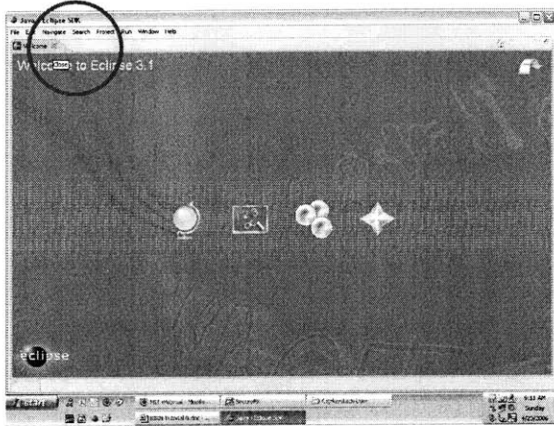
- Download Haystack-User.zip from <http://web.mit.edu/sumudu/www/haystack/download>
- Haystack-User.zip is approximately 200MB in size. It should take you around 5 minutes to download the file using a high-speed connection.

2. Install Haystack Evaluation Version

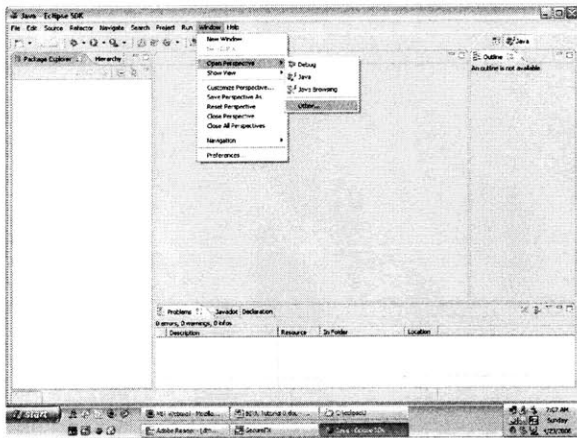
Follow the instructions given below to unzip Haystack-User into C:/Program Files

- Double-click on Haystack-User.zip
- Copy the folder Haystack-User
- Navigate to C:/Program Files and paste. Wait until the unzipping completes.

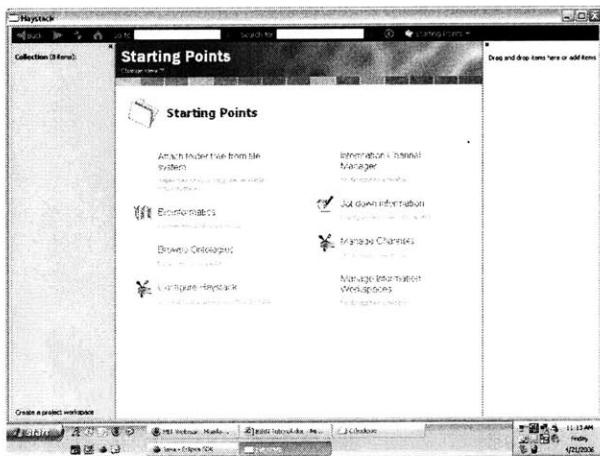
- Double-click on eclipse.exe in the Haystack-User directory
- Close the Welcome screen



- Select from the menu: Window>Open Perspective>Other...
- Select Haystack Full Screen



- The Haystack Starting Points screen:

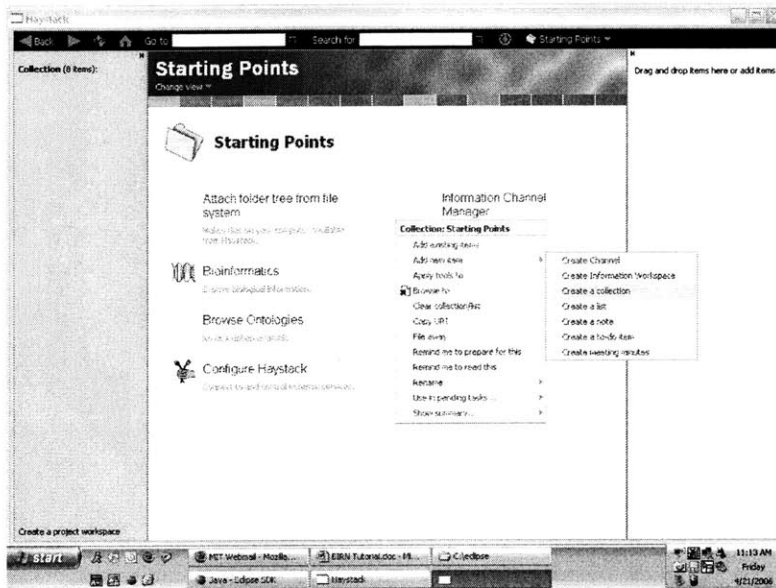


The remaining tasks in Part 1 are simple tasks designed to help you get used to the Haystack environment.

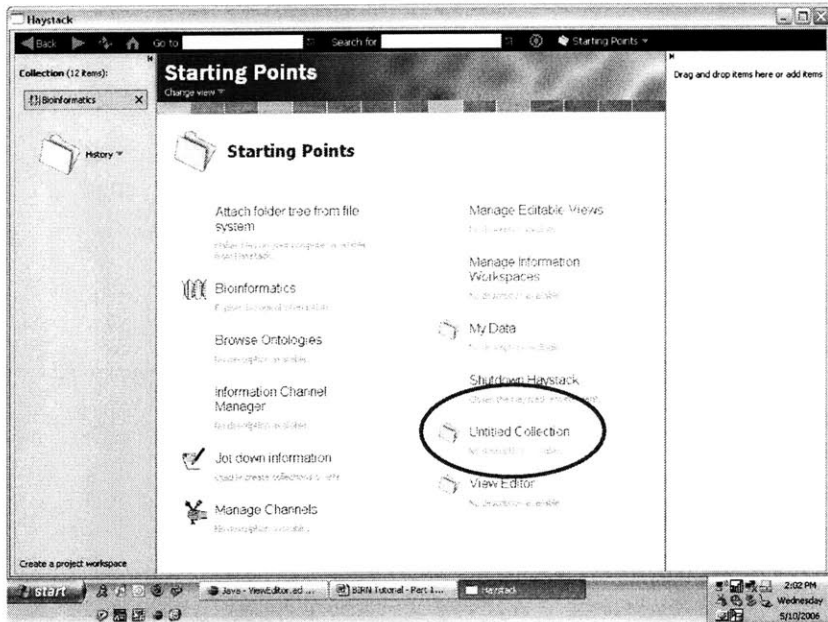
3. Create a new collection

A collection in Haystack is similar to folder in Windows. However, it is more powerful in that you can organize many different types of resources in it. For instance you can add operations, data, view information, queries, and other collections into a collection.

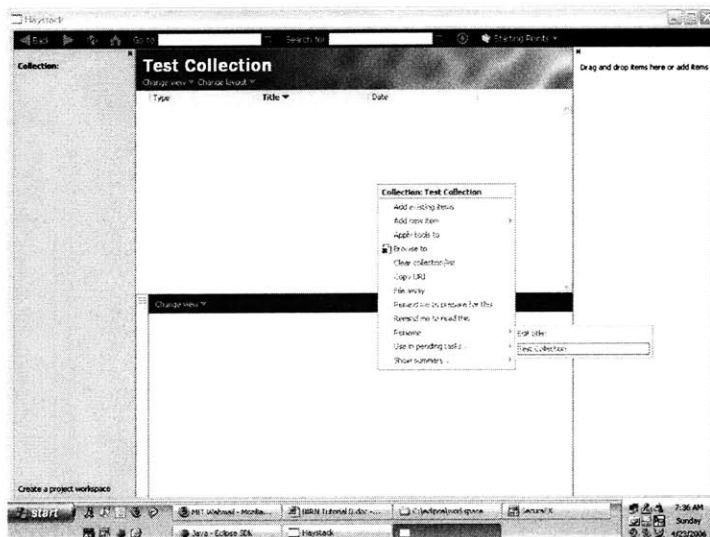
- Right-click on the middle pane
- In the *Collection: Starting Points* menu, mouse over *Add new item*
- Select *Create a collection*



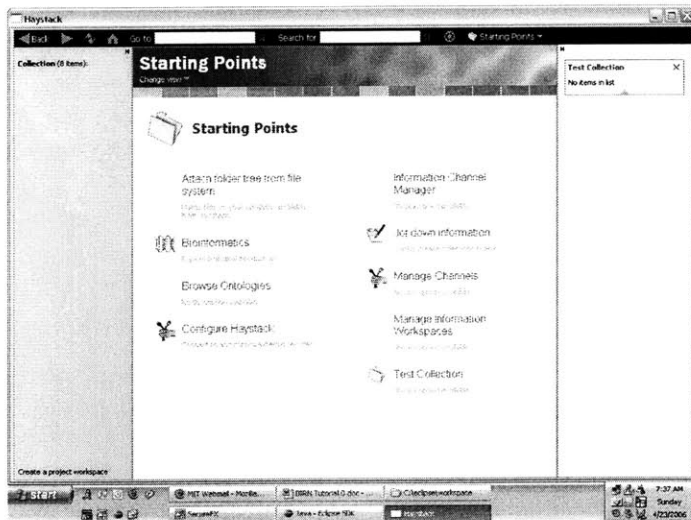
- Click on *Untitled Collection*



- Right-click on middle pane
- Mouse over *Rename*
- Type in new name (Test Collection)



- Go back to *Starting Points* (using the *Back* button)
- Drag and drop *Test Collection* to the right pane



4. Add items to Test Collection

Information can be organized in Haystack using tools like collections, quick-links, and favorites.

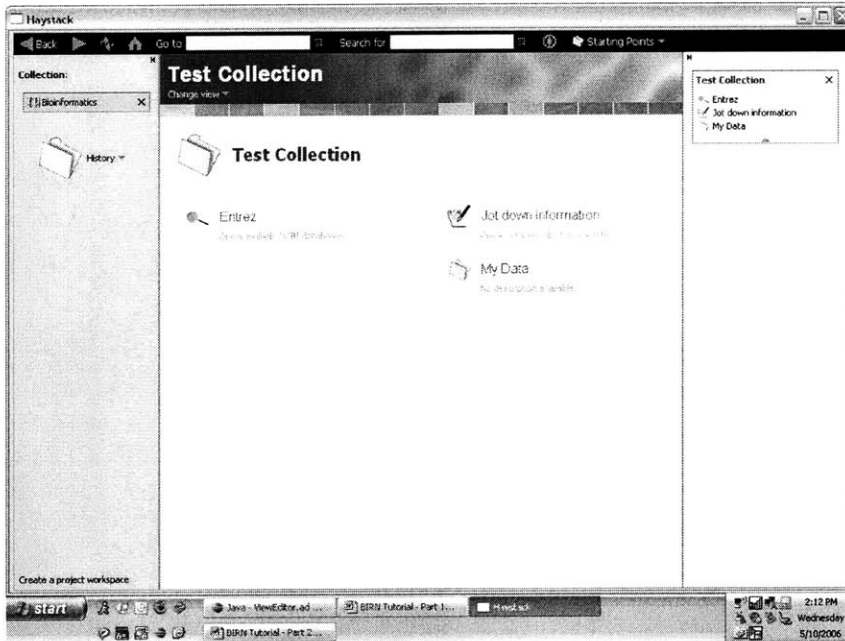
- Drag and drop the *Jot down information* operation to the *Test Collection* folder in *Starting Points*
- Drag and drop the *My Data* collection to the *Test Collection* slide on the right pane
- Click on *Bioinformatics*
- Drag and drop *Entrez* to the *Test Collection* slide on the right pane



5. Change views

Any information resource can be viewed in multiple ways in Haystack, depending on the task at hand or individual user preference. A "view" is a particular way of looking at a certain information resource.

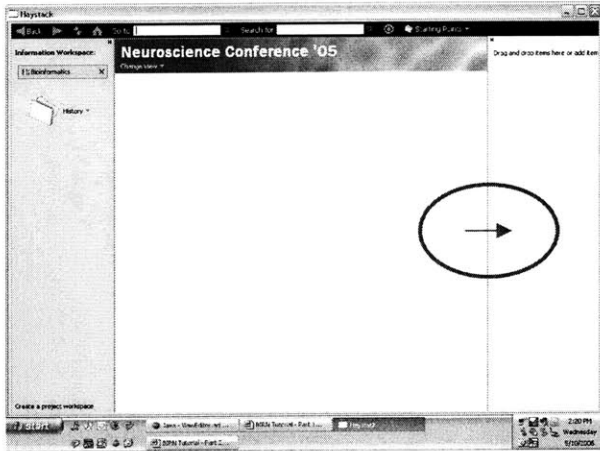
- Click on *Test Collection* on the right pane
- Currently the items in this collection are shown in a tabular view. Click on *Change view* in the title bar at the top of the middle pane to see other visualization options
- Select *Browse*



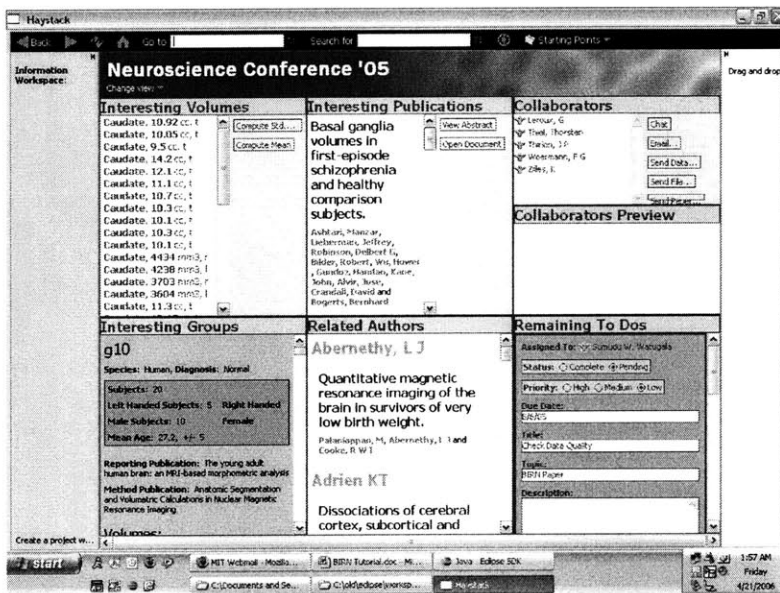
6. View example information workspace

A workspace is an environment that contains all the resources necessary to complete a particular task. A sample workspace in Haystack is the "Neuroscience Conference" workspace, created by a user to help write a paper for a conference.

- Go back to *Starting Points* using the *Back* button
- Click on *Manage Information Workspaces*
- Click on *Neuroscience Conference '05*
- Select *Change view > Show Information Workspace Usage View*
- Increase the size of the middle Haystack pane by dragging one of the edge lines out
- Wait a couple of seconds for loading to complete



- The workspace contains views of the data relating to the paper the user is working on, in addition to links to information about collaborators and related authors.



- End of Part 1 of User Study -

Part 2 - Tasks

Scenario:

You have recently started working on a new type of data that is related to your research on the human brain. You are working with data sets from different groups. Right now the data is located in the *My Data* collection in your *Starting Points*.

1. Organizing your data

You would like to have easy access to the data you are currently working on while you do your other tasks.

- Drag *My Data* to the right pane
- Click on the *Bioinformatics* collection on *Starting Points*

You would like to have all your research-related information in one collection, *Bioinformatics*

- Drag *My Data* into the *Bioinformatics* collection

Now you have your data linked from 2 different collections.



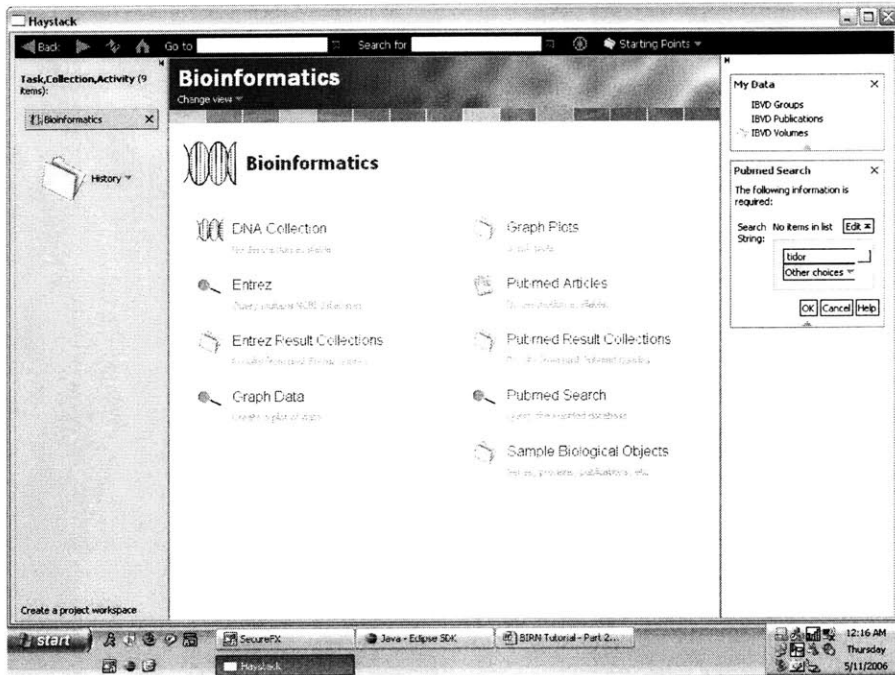
[Please fill out Section 1 of the User Feedback Form]

2. Query PubMed on relevant papers and manage results

- Right-click on the *PubMed Search* operation (in the *Bioinformatics* collection)
- In the *Operation: PubMed Search* menu, select *Perform Operation*

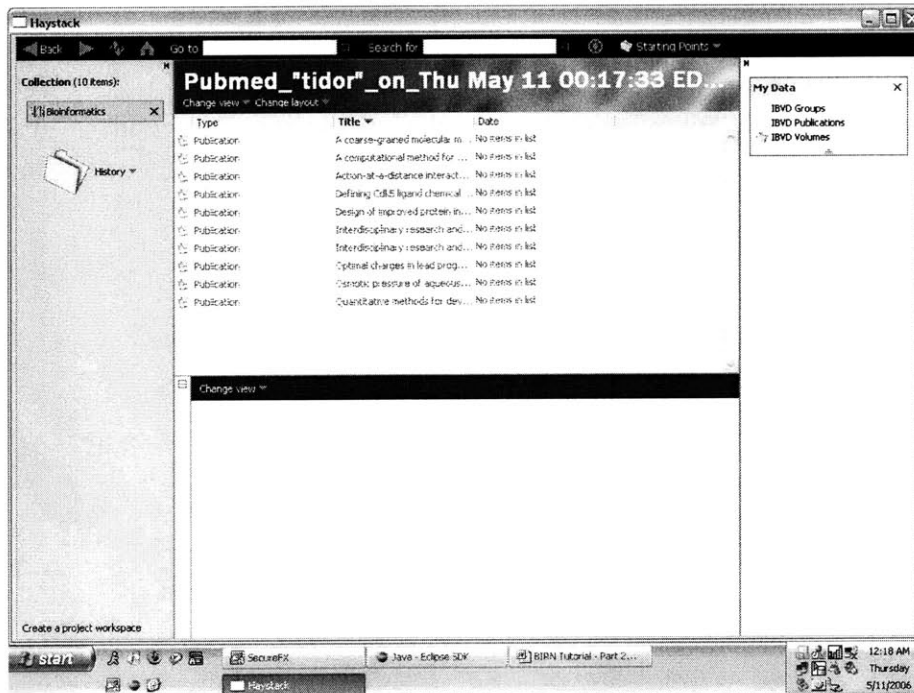


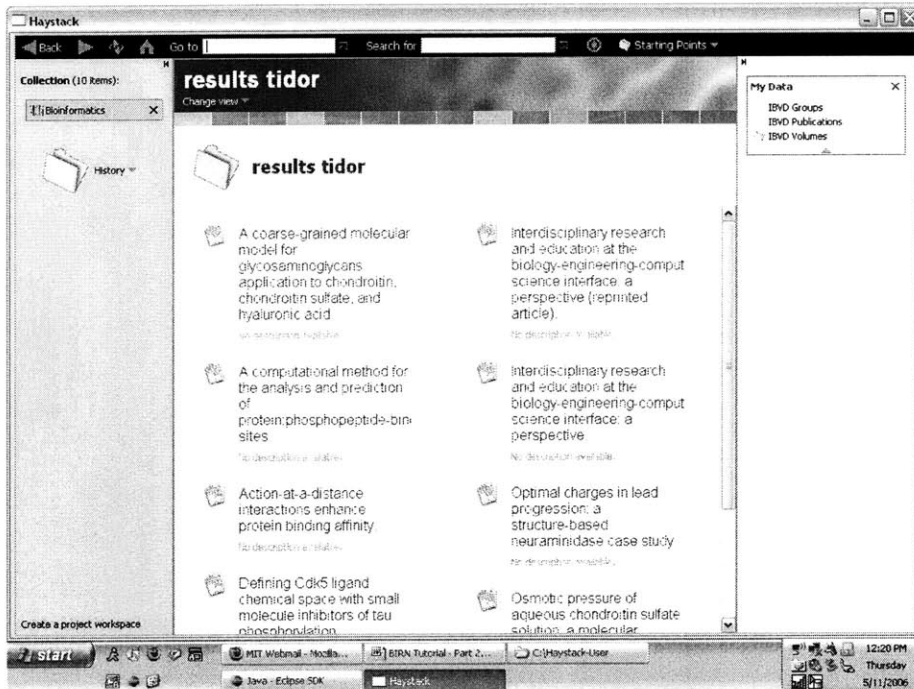
- Click *Edit* and type in your search string (tidor)
- Press Enter
- Click *Ok* and wait for the query to complete



A collection of results is displayed

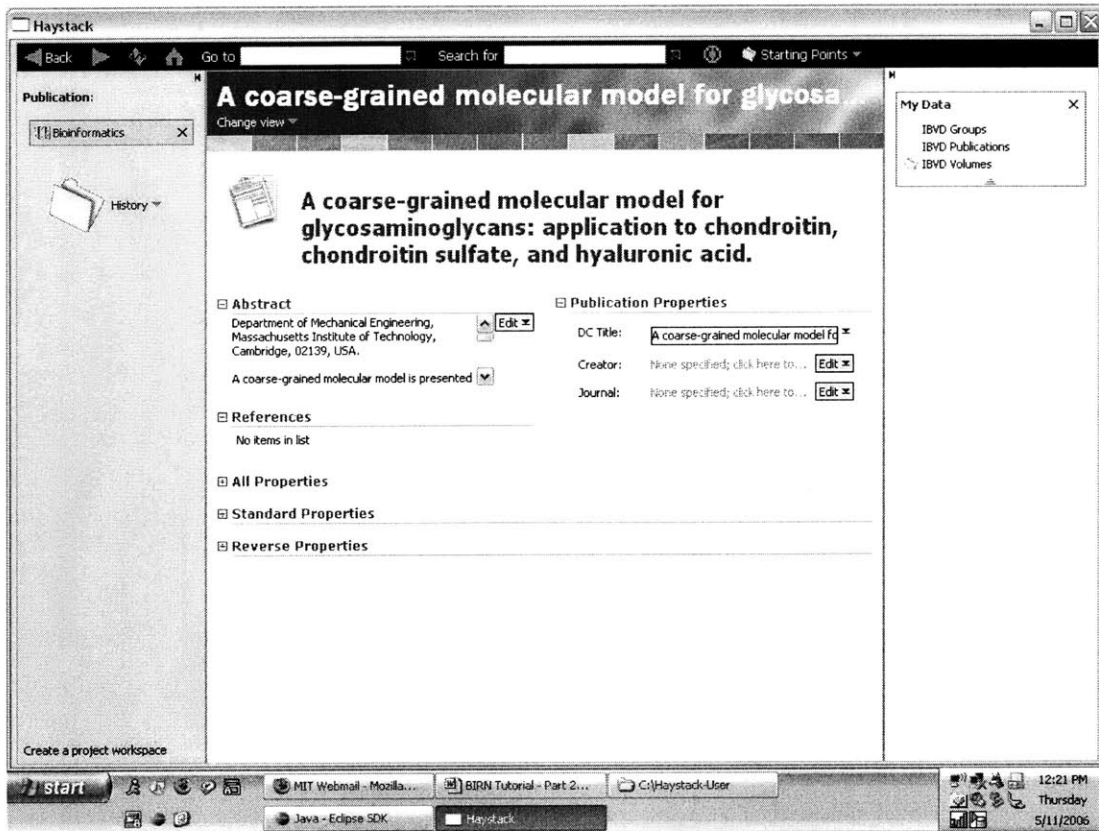
- Change the name of this collection to *Results for tidor*
- Change the view of the collection to *Browse view*



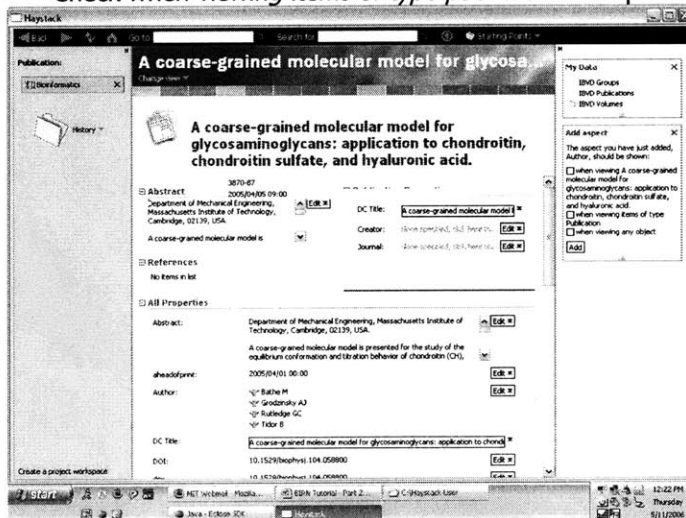


- Click on the title of the first PubMed result to view it in more detail

You would like to modify the way publications are displayed. Let's personalize the current view.



- Open the *All Properties* tab.
- Drag *Author* and drop on the white space above *All Properties*
- Check *when viewing items of type publication* and press *Add*



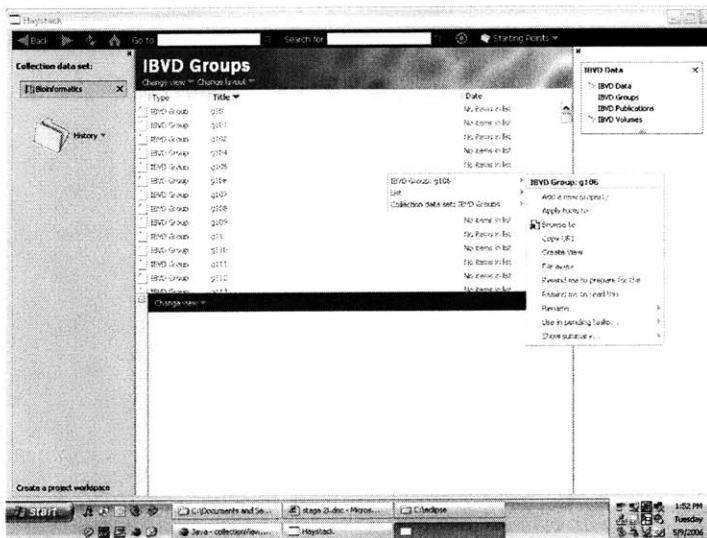
- Close the *All Properties* tab

Now all current and future PubMed results can be viewed according to this modified layout.

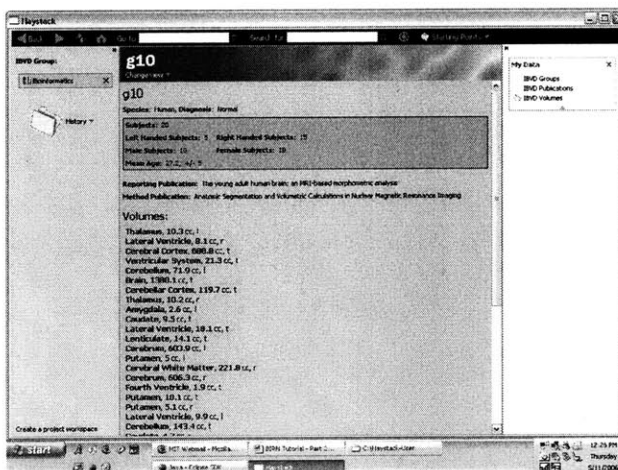
- Navigate back to *Bioinformatics* using the *Back* button

3. Viewing different types of data in different ways

- Click on *IBVD Groups* on the slide on the right pane



- Select a group (g10)
- Click on *Change view* and select *All information*
- Open the *All Properties* tab



[Please fill out Section 2 and 3 of the User Feedback Form]

4. Define relationships between your new and existing information

- Go *IBVD Publication* in *My Data* collection
- Click on any IBVD publication
- Right-click and mouse over *IBVD Publication:...*
- Select get paper from PubMed

This retrieves data about the paper from PubMed and links to two sets of information for future reference

[Please fill out Section 4 of the User Feedback Form]

Suggestions for additional exploration

In alternate versions of Haystack it is possible for the user to create more advanced personalized views and workspaces.

- Look at the examples below and comment on how this functionality could be useful. Describe what sort of views and workspaces would be helpful in your research work. How and what would you like to represent in a personalized way? Please describe your ideas in the Reflection section of the User Feedback form.

An example personalized view of gene data

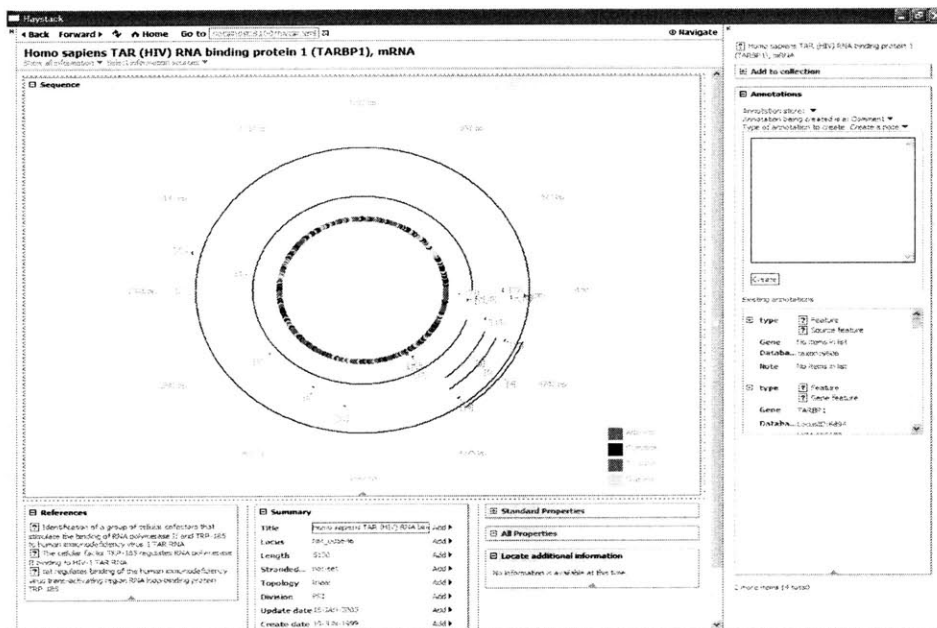


Figure 1: A view of an mRNA sequence in BioHaystack that shows selected properties, a graphical representation, and operations that can be run on it

An example personalized workspace:

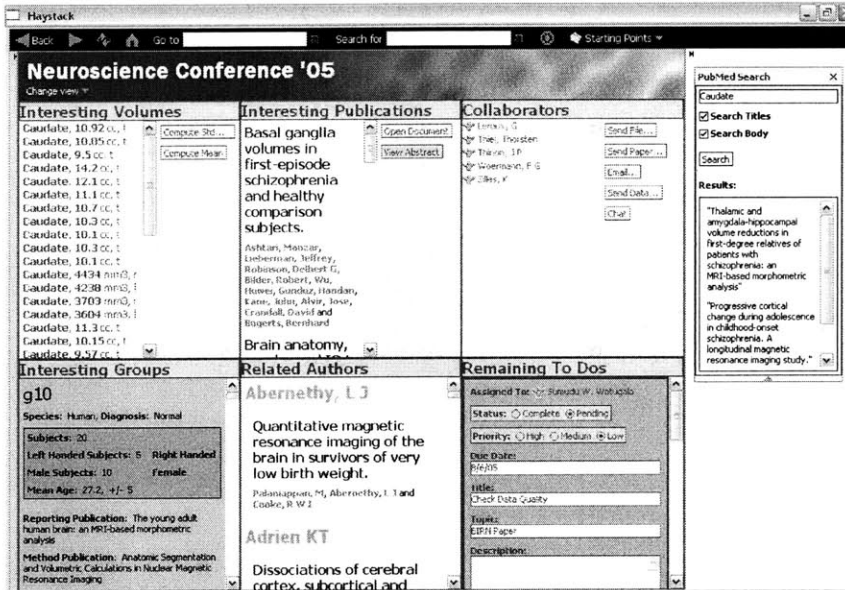


Figure 2: A workspace that shows everything the user needs to perform a certain task; in this case the task of writing a paper

An example view that shows relationships between different data objects

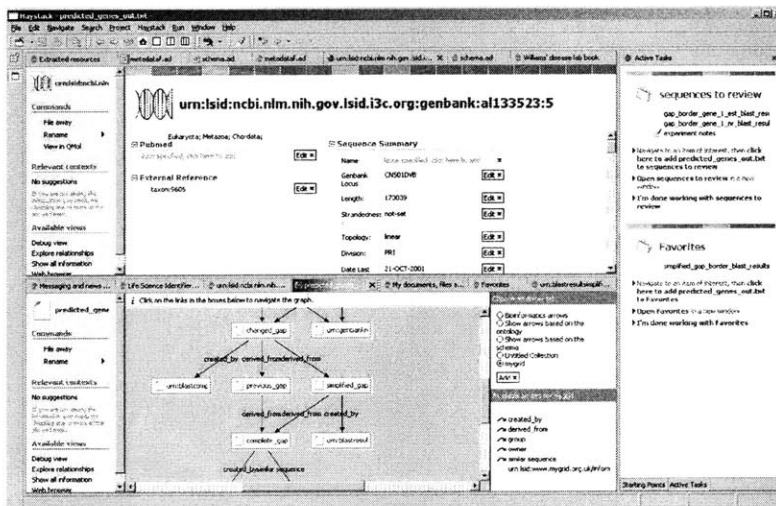


Figure 3: A view of a gene in BioHaystack that shows some of its attributes and its relations to other resources

[Please fill out the Reflections section of the User Feedback Form]

- End of Part 2 of User Study -

Appendix C – User Study Feedback Form

Please answer the following questions after completing the User Study Instructions and return to Sumudu W. Watugala before May 12th, 2006. Thank you for your cooperation.

Background

- 1) Name (Optional; for the purposes of following up on your feedback): _____
- 2) Age: _____
- 3) Gender: F M
- 4) Educational background (highest/current degree): _____
- 5) What is your field of research (neuroscience, bioinformatics, protein structure analysis, etc.)?

- 6) How long have you been doing research in your field?
 <1 year 1-2 years 2-5 years >5 years
- 7) How long have you used Microsoft Windows?
 <1 year 1-2 years 2-5 years >5 years
- 8) How long have you used the World Wide Web?
 <1 year 1-2 years 2-5 years >5 years
- 9) Have you heard about the Semantic Web prior to participating in this study? Yes No
If yes, in what context? _____

- 10) Have you used Haystack prior to this user study? Yes No
If yes, in what context? _____

Section 1

11) Annotating and managing your data so that you can organize your thoughts and find important information at a later date.

g) How often have you needed to perform similar tasks for your work outside of this user study?

- Never Rarely Sometimes Often

h) How do you currently perform such a task? What software, if any, do you use?

i) Describe one way in which Haystack is better for similar tasks than the current method.

j) Describe one drawback of Haystack compared to the current method when used to perform similar tasks.

k) Drawing from your experience from the user study, indicate which method was better for the task.

- Haystack Current method Both the same

l) If one method was better, why or how?

m) Viewing all application windows necessary to complete a task on the screen at once

a. How easy was it to perform the task in Haystack (1-difficult to 5-very easy)?

1 2 3 4 5 N/A

b. How useful was the feature (1-not useful to 5-very useful)?

1 2 3 4 5 N/A

c. How likely are you to do a similar task in the future (1-not likely to 5-very likely)?

1 2 3 4 5 N/A

d. How often do you need to do a similar task for your work (1-rarely to 5-very often)?

1 2 3 4 5 N/A

[Please continue on to Task 2 in the User Study Instructions]

Section 2

12) Querying databases related to your research; annotating and organizing results from your search in a way that will be easy to retrieve in the future.

a) How often have you needed to perform similar tasks for your work outside of this user study?

- Never Rarely Sometimes Often

b) How do you currently perform such a task? What software, if any, do you use?

c) Describe one way in which Haystack is better for similar tasks than the current method.

d) Describe one drawback of Haystack compared to the current method when used to perform similar tasks.

e) Drawing from your experience from the user study, indicate which method was better for the task.

- Haystack Current method Both the same

f) If one method was better, why or how?

g) Querying a database and annotating and sorting the results

- a. How easy was it to perform the task in Haystack (1-difficult to 5-very easy)?

1 2 3 4 5 N/A

b. How useful was the feature (1-not useful to 5-very useful)?

1 2 3 4 5 N/A

c. How likely are you to do a similar task in the future (1-not likely to 5-very likely)?

1 2 3 4 5 N/A

d. How often do you need to do a similar task for your work (1-rarely to 5-very often)?

1 2 3 4 5 N/A

Section 3

13) Visualizing your information in multiple ways, depending on the task at hand

a) How often have you needed to perform similar tasks for your work outside of this user study?

Never Rarely Sometimes Often

b) How do you currently perform such a task? What software, if any, do you use?

c) Describe one way in which Haystack is better for similar tasks than the current method

d) Describe one drawback of Haystack compared to the current method when used to perform similar tasks.

e) Drawing from your experience from the user study, indicate which method was better for the task.

Haystack Current method Both the same

f) If one method was better, why or how?

g) Visualizing data of the same type in different views

e. How easy was it to perform the task in Haystack (1-difficult to 5-very easy)?

1 2 3 4 5 N/A

f. How useful was having access to this feature (1-not useful to 5-very useful)?

1 2 3 4 5 N/A

g. How likely are you to do a similar task in the future (1-not likely to 5-very likely)?

1 2 3 4 5 N/A

h. How often do you need to do a similar task for your work (1-rarely to 5-very often)?

1 2 3 4 5 N/A

h) Creating a new way of visualizing data of a certain type

a. How easy was it to perform the task in Haystack (1-difficult to 5-very easy)?

1 2 3 4 5 N/A

b. How useful was the feature (1-not useful to 5-very useful)?

1 2 3 4 5 N/A

c. How likely are you to do a similar task in the future (1-not likely to 5-very likely)?

1 2 3 4 5 N/A

d. How often do you need to do a similar task for your work (1-rarely to 5-very often)?

1 2 3 4 5 N/A

[Please continue on to Task 4 in the User Study Instructions]

Section 4

14) Defining and exploring relationships between different data sets

a) How often have you needed to perform similar tasks for your work outside of this user study?

- Never Rarely Sometimes Often

b) How do you currently perform such a task? What software, if any, do you use?

c) Describe one way in which Haystack is better for similar tasks than the current method

d) Describe one drawback of Haystack compared to the current method when used to perform similar tasks.

e) Drawing from your experience from the user study, indicate which method was better for the task.

- Haystack Current method Both the same

f) If one method was better, why or how?

g) Visualizing data of the same type in different views

- a. How easy was it to perform the task in Haystack (1-difficult to 5-very easy)?

1 2 3 4 5 N/A

b. How useful was the feature (1-not useful to 5-very useful)?

1 2 3 4 5 N/A

c. How likely are you to do a similar task in the future (1-not likely to 5-very likely)?

1 2 3 4 5 N/A

d. How often do you need to do a similar task for your work (1-rarely to 5-very often)?

1 2 3 4 5 N/A

[Please continue on to the next section of the User Study Instructions]

Reflection

- 15) What sort of views would you like to create to visualize your information in different ways? For which data types? Why is this view preferable to what you currently have? Please use the space below to draw what you have in mind.

- 16) What sort of workspaces would you like to create to facilitate your tasks? When and why would you use these workspaces? Please use the space below to draw what you have in mind.

17) What tasks do you think could be made easier by using an *improved* version of Haystack?
How or why?

18) Which other tasks do you think could be done in Haystack that would facilitate your research?

19) Other comments and suggestions:

- End -

Appendix D – User Responses

Key for Table D-1:

Viewing all application windows necessary to complete a task on the screen at once (11g)

Querying a database and annotating and sorting the results (12g)

Visualizing data of the same type in different views (13g)

Creating a new way of visualizing data of a certain type (13h)

Making persistent connections between different data sets for future use (14g)

Table D-1: Individual ratings on tasks for ease of use, usefulness of feature, likelihood of repeating similar tasks, and the estimated frequency of repetition

Question	11g				12g				13g				13h				14g			
	a	b	c	d	a	b	c	d	a	b	c	d	a	b	c	d	a	b	c	d
1	4	3	4	4	4	5	5	4	3	4	4	3	3	4	3	3	3	4	4	3
2	4	4	5	5	3	4	5	5	4	4	4	4	3	4	3	3	-	-	-	-
3	4	3	5	4	3	-	4	3	-	5	5	4	5	5	3	3	5	5	4	3
4	5	4	4	3	5	5	5	5	4	4	4	4	4	4	4	4	4	4	4	4
5	5	4	3	2	5	5	5	4	5	4	4	2	4	3	2	2	5	5	5	4
6	5	4	5	5	5	4	4	4	5	4	5	4	4	4	3	3	4	5	5	5

a – How easy was it to perform the task in Haystack? (1-difficult to 5-very easy)
b – How useful was the feature? (1-not useful to 5-very useful)
c – How likely are you to do a similar task in the future? (1-not likely to 5-very likely)
d – How often do you need to do a similar task for your work? (1-rarely to 5-very often)

References

- [1] Quan, D., Huynh, D. and Karger, D. Haystack: A Platform for Authoring End User Semantic Web Applications. 2nd International Semantic Web Conference, 2003.
- [2] Quan, D. Designing End User Information Environments Based On Semi-structured Data Models. PhD Thesis, MIT, 2003.
- [3] Karger, D. R., Quan, D. Prerequisites for a Personalizable User Interface. Workshop on Behavior-based User Interface Customization, Intelligent User Interfaces (IUI) 2004.
- [4] Huynh, D. Haystack's User Interface Framework: Tutorial and Reference: <http://haystack.lcs.mit.edu/documentation/ui.pdf>.
- [5] Quan, D., – Invited Talks: <http://theory.csail.mit.edu/~dquan/www2004-biohaystack.ppt>.
- [6] Quan, D., Karger, D., Huynh, D. RDF Authoring Environments for End Users. International Workshop on Semantic Web Foundations and Application Technologies (SWFAT 2003).
- [7] Berners-Lee, T., Hendler, J., Lassila, O. The Semantic Web, Scientific American, May 2001.
- [8] Huynh, D., Karger, D., Quan, D. Haystack: A platform for creating, organizing, and visualizing information using RDF. In Semantic Web Workshop at WWW2002, Hawaii, May 2002.
- [9] Quan, D., Bakshi, K., Huynh, D., Karger, D. User interfaces for supporting multiple categorization. In *INTERACT: 9th IFIP International Conference on Human Computer Interaction*, Zurich, September 2003. International Federation for Information Processing.
- [10] Quan, D., Karger D. Haystack: A platform for authoring end-user semantic web applications. In *Proceedings of the International Semantic Web Conference*, 2003.
- [11] Quan, D. A. and Karger, D. R. 2004. How to make a semantic web browser. In Proceedings of the 13th international Conference on World Wide Web (New York, NY, USA, May 17 - 20, 2004). WWW '04. ACM Press, New York, NY, 255-265.

- [12] Kelly, D., Teevan, J. "Understanding what works: Evaluating PIM tools." To appear in William Jones and Jaime Teevan (Eds.), *Personal Information Management*. Seattle: University of Washington Press, 2007 (In preparation).
- [13] Huynh, D. A User Interface Framework for Supporting Information Management Tasks in Haystack. Master's Thesis, MIT, May 2003.
- [14] Bakshi K. Tools for End-User Creation and Customization of Interfaces for Information Management Tasks. Master's thesis, M.I.T., May 2004.
- [15] Quan, D., Huynh, D., Karger, D., Miller, R. User Interface Continuations, UIST 2003.
- [16] Quan, D., *Metadata Programming in Adenine*, February 2003.
<http://haystack.csail.mit.edu/documentation/adenine.pdf>.
- [17] Palmiter, S., Lynch, G., Day, J., Geist, M., and Rhoads, B. 2005. Focus on the individual: the future of web-based product support. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems* (Portland, OR, USA, April 02 - 07, 2005). CHI '05. ACM Press, NY, 2136-2137.
- [18] Dolog, P., Henze, N., Nejd, W., and Sintek, M. 2004. Personalization in distributed e-learning environments. In *Proc. of the 13th international World Wide Web Conference on Alternate Track Papers & Posters* (NY, USA, May 19 - 21, 2004). WWW Alt. '04. ACM Press, NY, 170-179.
- [19] Peyton, L. 2003. Measuring and managing the effectiveness of personalization. In *Proceedings of the 5th international Conference on Electronic Commerce* (Pittsburgh, Pennsylvania, September 30 - October 03, 2003). ICEC '03, vol. 50. ACM Press, New York, NY, 220-224.
- [20] Massar, J. P., Travers, M., Elhai, J., Shrager, J. BioLingua: a programmable knowledge environment for biologists. *Bioinformatics Advance Access* published on January 15, 2005. *Bioinformatics* 21: 199-207.
- [21] Stevens, R. D., Robinson A. J., Goble, C. A.. myGrid: personalised bioinformatics on the information grid. *Bioinformatics* 19: i302-i304.
- [22] Lord, P., Bechhofer, S., Wilkinson, M.D., Schiltz, G., Gessler, D., Hull, D., Goble, C., Stein, L. (2004), Applying semantic web services to Bioinformatics: Experiences gained, lessons learnt, ISWC 2004.

- [23] Clark, T., Martin, S., Liefeld, T. Globally Distributed Object Identification for Biological Knowledgebases, Briefings in Bioinformatics 5.1:59-70, March 1, 2004.
- [24] Hendler, J. Science and the Semantic Web, Science 299: 520-521, 2003.
- [25] Neumann, E. A Life Science Semantic Web: Are We There Yet? Sci. STKE, Vol. 2005, Issue 283, 10 May 2005.
- [26] Adar, E., Karger, D., and Stein, L. A. 1999. Haystack: per-user information environments. In Proceedings of the Eighth international Conference on information and Knowledge Management (Kansas City, Missouri, United States, November 02 - 06, 1999). S. Gauch, Ed. CIKM '99. ACM Press, New York, NY, 413-422.
- [27] Sinha, V., Karger, D. Magnet: Supporting navigation in semistructured data environments. SIGMOD 2005.
- [28] Zhao, J., Wroe, C., Goble, C., Stevens, R., Quan, D., Greenwood, M., *Using Semantic Web Technologies for Representing e-Science Provenance* in Proc 3rd International Semantic Web Conference ISWC2004, Hiroshima, Japan, 9-11 Nov 2004, Springer LNCS 3298.
- [29] Cheung K.H., Yip K.Y., Smith A., deKnikker R., Masiar A., Gerstein M. *YeastHub: a semantic web use case for integrating data in the life sciences domain* (2005) *Bioinformatics* 21 Suppl 1: i85-i96.
- [30] Clark T., Martin S., Liefeld T. *Globally Distributed Object Identification for Biological Knowledgebases* Briefings in Bioinformatics 5.1:59-70, March 1, 2004.
- [31] Goble C. A., De Roure D., Shadbolt N.R., Fernandes A. A. A. *Enhancing Services and Applications with Knowledge and Semantics* in The Grid: Blueprint for a New Computing Infrastructure Second Edition (eds. I Foster and C Kesselman), Morgan Kaufman 2003.
- [32] Lord, P., Bechhofer, S., Wilkinson, M., Schiltz, G., Gessler, D., Goble, C., Lincoln Stein, Duncan Hull. *Applying semantic web services to bioinformatics: Experiences gained, lessons learnt.* in Proc 3rd International Semantic Web Conference ISWC2004, Hiroshima, Japan, 9-11 Nov 2004 , Springer LNCS 3298.
- [33] Haystack Project. <http://haystack.csail.mit.edu/home.html>.
- [34] Resource Description Framework (RDF). <http://www.w3.org/RDF/>.

- [35] Extensible Markup Language (XML). <http://www.w3.org/XML/>.
- [36] Web Services Activity. <http://www.w3.org/2002/ws/>.
- [37] Semantic Web. <http://www.w3.org/2001/sw/>.
- [38] NIH working definition of bioinformatics and computational biology. <http://www.bisti.nih.gov/CompuBioDef.pdf>.
- [39] Life Sciences Identifiers final adopted specification. <http://www.omg.org/docs/dtc/04-05-01.pdf>.
- [40] The Eclipse Project. <http://www.eclipse.org/>.
- [41] Haystack developer documentation site. <http://haystack.lcs.mit.edu/developers/>.
- [42] W3C. "XML Path language (XPath) Version 1.0," 1999.
- [43] BioPathways Consortium. <http://lsid.biopathways.org/>.
- [44] BIRNing issues - Biomedical Informatics Research Network Newsletter, February 2005, Vol.3, Issue 2. <http://www.nbirn.net/AU/HotTopics/HotTopics.htm>.
- [45] BIRN Coordinating Center: <http://www.nbirn.net/TestBeds/CoordinatingCenter/index.htm>.
- [46] Brain Morphometry Test Bed (mBIRN). <http://www.nbirn.net/TestBeds/Morphometry/index.htm>.
- [47] GoPubMed, Ontology-based Literature Search. <http://www.gopubmed.org/>.
- [48] The Gene Ontology. <http://www.geneontology.org/>.
- [49] 3D Slicer. <http://www.slicer.org/>.
- [50] NeuroLens. <http://www.neurolens.org/>.
- [51] The Tidor Group. <http://web.mit.edu/tidor/www/>.
- [52] Semantic Web Application in Neuromedicine (SWAN). <http://swan.mindinformatics.org>.
- [53] SWAN prototype. <http://www.mindinformatics.org:8081/swan>.
- [54] Internet Brain Volume Database. <http://www.cma.mgh.harvard.edu/ibvd/index.php>
- [55] Wikipedia. <http://en.wikipedia.org/>.
- [56] Windows Live Beta. <http://www.live.com/>.
- [57] Entrez, The Life Sciences Search Engine. <http://www.ncbi.nih.gov/Entrez/>
- [58] Amazon.com. <http://www.amazon.com/>.
- [59] Google Personal Homepage. <http://www.google.com/ig>.