CELO: A System for Efficiently Building Informatics Solutions to Manage Biomedical Research Data

Christine Fong

A thesis submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2005

Program Authorized to Offer Degree:
Department of Medical Education and Biomedical Informatics

University of Washington

Graduate School

This is to certify that I have examined this copy of a master’s thesis by

Christine Fong

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

____
James F. Brinkley

____
John I. Clark

____
Peter Tarczy-Hornoch

Date:_______________________________

Signature_______________________________

Date___________________________________

University of Washington

Abstract

CELO: A Cost-Effective System for Efficiently Building
Informatics Solutions to Manage Biomedical Research Data

Christine Fong

Chair of the Supervisory Committee:
Research Professor James F. Brinkley
Department of Biological Structure

Traditional data management methods are unable to sufficiently support growing trends in biomedical research such as collection of larger data sets, use of diverse data types, and sharing of data among multiple laboratories. Although many technologies are readily available to help laboratories build data management solutions, many laboratories are not taking advantage of them. This may be due to hardware and software costs, the need for an informaticist to build customized solutions, and long development times.

Several systems already exist which attempt to address the informatics needs of biomedical researchers. A review of these systems has revealed the benefits and drawbacks of various system design approaches, and has helped us to identify a set of core requirements for a system that will successfully serve the biomedical research community. In consideration of these requirements, we developed the Customizable Electronic Laboratory Online (CELO) system to help laboratories efficiently build cost-effective informatics solutions. CELO automatically creates a generic database and web interface for laboratories that submit a simple web registration form. Researchers can then build their own customized data management systems using web-based features such as configurable user permissions, customizable user interfaces, support for multimedia files, and templates for defining research data representations.

An evaluation of the CELO system has demonstrated its ability to efficiently create customized solutions for research laboratories with basic data management needs. The evaluation has also highlighted areas in which CELO can be improved and has elucidated potential research problems that may be of interest to the biomedical informatics field.

Table of Contents

Page

List of Figures

Figure Number Page

List of Tables

Table Number Page

Chapter 1: Introduction

1.1 Data management needs in biomedical research labs

Data management is a critical aspect of biomedical research[1-3]. Data management involves recording, storing, organizing, retrieving and visualizing data. Different methods for managing research data can affect the ease and efficiency with which research is performed and can also play a role in the quality of research[4-8]. The nature of recent research has led to a trend towards the collection of larger and larger data sets that are becoming more difficult to manage. The initial sequencing of the human genome, for example, has resulted in an explosion of genomic studies that generate a high volume of information such as gene mapping or gene expression data[9]. Recent technologies such as digital video and photography have also enabled researchers to rapidly perform cost effective, large scale experiments that require analysis of thousands of images[10]. Some neuroscience studies that contribute to the goal of mapping the approximately $100x10^9$ neurons of the human brain also result in enormous data sets[11]. Traditional methods for managing data, such as lab books, typewritten documents, and spreadsheets created using software such as Microsoft Excel, are becoming insufficient for handling this volume of data. These methods lack querying capabilities that enable researchers to efficiently find and retrieve particular data items. Effectively organizing such large amounts of data for analysis also becomes nearly impossible using traditional methods[12-18].

Digital images and movies are among several other types of computerized formats that are more commonly being collected as research data. Software applications built for specific research areas result in a diverse set of very domain specific data files. For example, the Visual Brain Mapper (VBM), an application developed specifically to visualize brain mapping data from neurosurgical studies, uses an input file specifying patient information and generates multiple brain map and model files[19]. In addition to these newer emerging types of data files, traditional computerized text documents and spreadsheets also continue to contribute to a typical biomedical research lab's data set. Management of all these diverse types of files is most typically performed through filesystem directory structures and file naming conventions devised by laboratory research members. Large collections of files, however, can be difficult to organize this way, presenting a challenge for researchers to find or compare specific files[20].

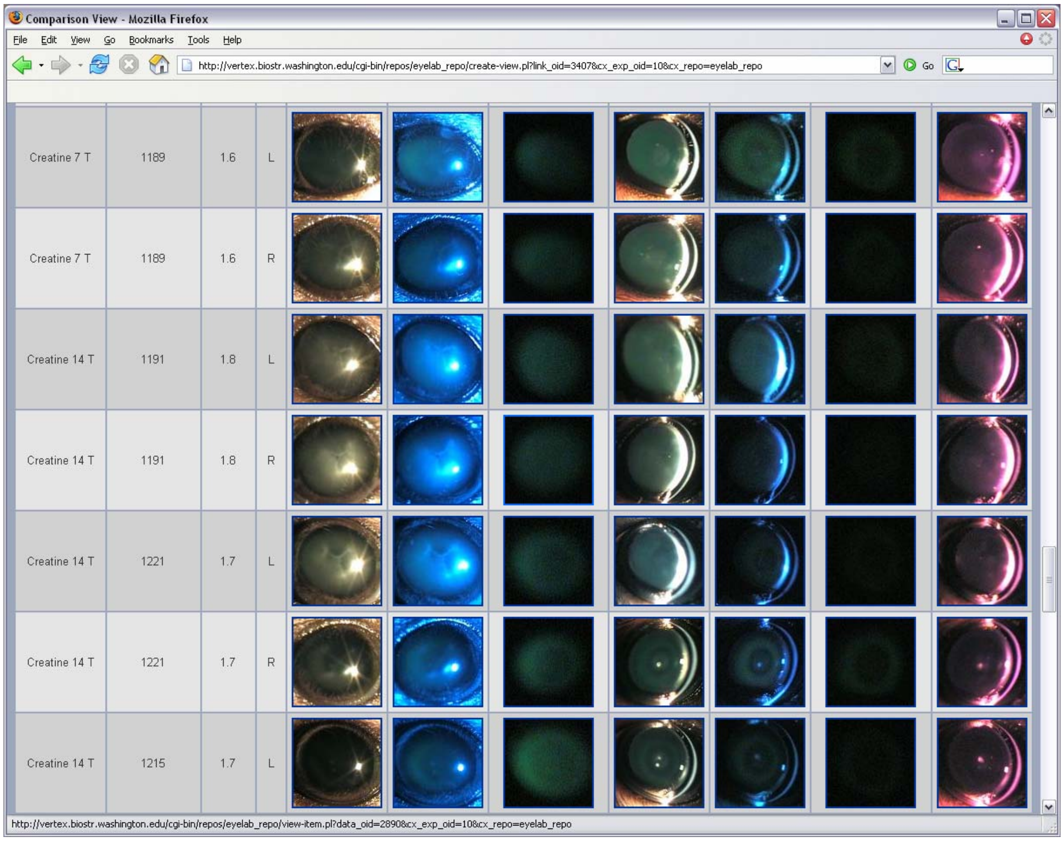
Another recent trend among biomedical research is the employment of large scale, multi-laboratory research efforts[6, 7, 11, 16, 21]. The research goal of mapping the human brain, for example, requires multidisciplinary expertise in the molecular, cellular and behavioral aspects of the neural system[11]. Collaborative research results in a growing need for methods that allow researchers to easily, efficiently and robustly share data remotely.

The informatics needs of today's biomedical research laboratories stem from trends in larger data sets, a growing diversity of computerized data types, and a need to share data for research collaborations. Fortunately, there are many technologies currently available that enable biomedical research labs to build informatics solutions that address these needs. As members of the University of Washington's Structural Informatics Group, we have had the opportunity to use these technologies to successfully build informatics solutions for multiple biomedical laboratories. Our experiences building data management systems for these research labs have helped us to identify some of the barriers labs may face when attempting to build their own database systems.

1.2 The barriers to building customized informatics solutions

Our observations from working with biomedical research labs have led us to recognize monetary and time costs as two major barriers that laboratories face when trying to build a customized informatics solution. Some labs have taken advantage of the readily available database and internet technologies to successfully build custom data management systems. These labs also often acknowledge cost as an important aspect to consider when implementing a system[4, 17]. We use our experiences building one such system to demonstrate why we believe time and money are such critical issues for labs in need of an informatics solution.

We worked closely with the members of a research lab in the University of Washington's Department of Biological Structure in order to assess their specific informatics needs and build a customized informatics solution. The lab, which we will refer to as the Eye Lab, studies characteristics of and factors affecting the development of cataracts. The lab performs experimental studies that generate thousands of images of the eyes of mice which need to be compared and analyzed. Previous data management methods involved storing thousands of image files into an operating system file folder structure and selecting filenames to specify details about each image, such as the mouse id, age, genetic information, and the eye (left or right) shown in the photo. Matrices of eye images for analysis of lens opacification patterns due to cataract were manually constructed using generic spreadsheet software and were time consuming to build and modify. In order to improve and expedite these data management processes, we built an image repository system that stores data files along with associated metadata that describes each eye image[22, 23]. We designed a web interface for querying the relational database that stores these data so that the Eye Lab researchers can easily find and retrieve particular images. We also designed a tool that enables the researchers to select image metadata constraints in order to automatically build image matrices similar to the spreadsheets they were previously creating by hand (Figure 1). The system has helped decrease the time required to organize and evaluate mouse eye image files. The image repository is accessible through a typical web browser, allowing the researchers to also easily share data with collaborating laboratories.



http://vertex.biostr.washington.edu/cgi-bin/repos/eyelab_repo/create-view.pl?link_oid=3407&cx_exp_oid=10&cx_repo=eyelab_repo

Creatine 7 T	1189	1.6	L
Creatine 7 T	1189	1.6	R
Creatine 14 T	1191	1.8	L
Creatine 14 T	1191	1.8	R
Creatine 14 T	1221	1.7	L
Creatine 14 T	1221	1.7	R
Creatine 14 T	1215	1.7	L

http://vertex.biostr.washington.edu/cgi-bin/repos/eyelab_repo/view-item.pl?data_oid=2890&cx_exp_oid=10&cx_repo=eyelab_repo

Figure 1. The Eye Lab Image Repository Image Matrix

The Eye Lab image repository was successfully built to facilitate data management and increase efficiency of data analysis. The cost incurred on the Eye Lab for building this system was minimal due to the unique collaboration with the Structural Informatics Group. We, from the Structural Informatics Group, used our own hardware and personnel to install and set up the necessary components of the image repository system. Our informatics skills were also required to develop custom scripts to create custom interfaces and functionality. We realize that many other research labs cannot afford a dedicated informaticist to build and maintain a customized data management system. Many labs may additionally be unable to invest in the hardware resources necessary to create such a system. Methods to reduce these hardware or personnel costs would help address the monetary barrier that labs may face when building custom informatics solutions.

We also observed the time cost of building the Eye Lab Image Repository, with the initial system completed in several weeks followed by several months of maintenence and improvements. The system development time might pose an issue for some labs with a pressing need for an informatics solution. We observed several areas, however, in which the development of the Eye Lab system was inefficient and could be improved. Firstly, the Eye Lab members spent valuable research time helping us understand the details of their research data so that we could design a system that would address their informatics needs. Time may have been saved if the researchers were provided with tools enabling them to design the system themselves[2, 13, 18]. We also wrote several custom scripts with only minor differences in order to generate user interfaces that satisfied the specific needs of the Eye Lab. Many of these user interfaces could have been easily captured using an automatically generated interface with simple configuration options[3, 13, 18]. We also created multiple database schemas that were very similar to each other to represent data for different experiments performed in the laboratory. This observation helped spark the idea of allowing researchers to reuse database schemas with the goal of increasing the efficiency with which such schemas are defined.

Observations from working with the Eye Lab have emphasized a demonstrated need to reduce the costs and the development time for building customized informatics solutions. We believe that a system designed to address time and money issues will help a greater number of laboratories to take advantage of the technologies available for creating informatics solutions. We have designed the Customizable Electronic Laboratory Online (CELO) system in response to the need for reducing monetary and time costs. The CELO design considers many of the observations we have made with our past experiences building laboratory data management systems, as well as several issues discussed in the literature regarding other custom systems.

1.3 CELO's approach to addressing the problem

Our goal for the CELO project was to provide biomedical laboratories with the tools necessary to inexpensively and efficiently build data management systems that meet their specific needs. In this section, we describe the approach we took when developing CELO and explain why we believe this approach will help us to achieve our goal. In our evaluation of CELO, we will assess how well the system actually meets our goal.

To help reduce the hardware and software costs that a laboratory must invest to create a system, CELO is designed to use a distributed resource model. With this model, one CELO system installation and set up, including a single database and web server, is shared among multiple laboratories. Informatics personnel must only administer this centralized system, thereby reducing the need for a dedicated informaticist for each lab. Each laboratory with access to the server can then take advantage of CELO's features in order generate an individualized laboratory system (Figure 2).

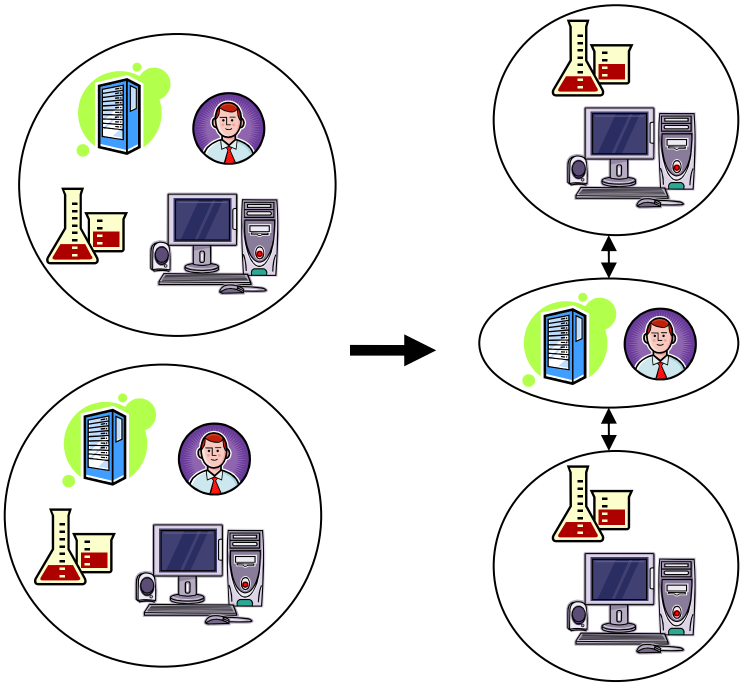


Figure 2. Distributed Resource Concept

Because research labs can register and build data management systems using CELO's web based interface, we believe time will be saved from having to install and set up a dedicated system. Since accessing the system simply requires a typical web browser, most users will automatically be able to utilize CELO without having to install any client software. We believe CELO can also help reduce development time by providing a set of generic system features that we believe will be useful for multiple laboratories. These generalizable features are determined by the needs of biomedical research labs that are emphasized in the literature, as well as needs highlighted by our own experiences working with labs. A mechanism for storing and organizing diverse file types, including multimedia data, on the system server helps laboratories manage the growing numbers of data files being produced for research. Flexible querying methods assist researchers with finding data and creating views for visualizing data. A permissions system facilitates data sharing among laboratories by allowing laboratories to control access to certain system functions. These examples are only a portion of the generalizable features that can be used by labs that have simply registered for a CELO-generated system online.

The CELO system design has also focused on making its features configurable in order to allow researchers to customize their systems based on specific needs. For example, CELO provides users with web based configuration tools for specifying graphical user interface customizations to best fit their research needs. The idea is that these tools will help lower development time by reducing the need for an informaticist to write custom scripts.

Our experiences building systems for research labs have also demonstrated that working with biomedical researchers to design the database schema for representing research data can be a time consuming task. To help increase efficiency with designing database schemas, we developed a template system. CELO templates specify pre-defined database schemas, as well as user interface details, and can be browsed and selected from in order to automatically generate a set of database tables that describe a given research area. For example, the Eye Lab could use a template to generate the database schema for an experiment studying various treatments for mice with cataracts. We believe that researchers without any database background can utilize the template system in order to design a database schema without the assistance of an informaticist. We expect that the templates will also foster database schema sharing, reuse, and consistency.

1.4 Contributions of this thesis

This chapter has introduced the growing need to provide tools for biomedical researchers that reduce the costs of laboratory data management systems and increase the efficiency with which they are created. This thesis discusses several of the informatics needs of the evolving biomedical research laboratory and reviews some of the existing systems that address these needs. We introduce a set of essential requirements for a system that we believe will more successfully address these needs based on this review of existing systems and our own personal experiences. This thesis introduces an approach to satisfy most of these requirements and details the architecture of the CELO system that implements this approach. We illustrate how a laboratory can easily use the CELO features to generate and configure a customized data management system.

An initial evaluation of CELO focuses on testing the hypothesis that users can use the web-based tools to efficiently build a system that meets their data management needs. To test this hypothesis, we used CELO to recreate three data management systems that we previously built. Our evaluation assessed CELO's ability to implement the major features required by the various laboratories and to compare the time needed to develop these features in the original system versus the CELO-generated system. We also created a system for a laboratory we had not previously worked with to help evaluate the generalizability of our system. The evaluation results demonstrate that the web-based tools such as the XML template system, configurable permissions system, customizable interfaces, and flexible querying mechanisms reduce the time needed to implement many features required by laboratories. Some essential system features, such as special formatting for displaying database items or query result lists, were not able to be captured using CELO's configuration tools, suggesting areas in which our system can be improved. Other critical system limitations highlighted by the evaluation include the lack of support for integrating custom features, or plugins, into the existing system and the inability to efficiently evolve research data representations while minimizing data loss. These limitations indicate that CELO is not a suitable solution for laboratories with more complex informatics needs. We conclude that the system is most valuable for laboratories that need a quick and inexpensive solution to perform basic data management tasks. This thesis not only demonstrates the potential value of the CELO system for such laboratories, but also generates ideas of future work for both our system and for the field of biomedical informatics in general.

Chapter 2: Background

The need for tools to facilitate the creation and maintenance of customized laboratory data management systems is well known[1, 2, 24]. Several informatics options already exist to assist biomedical research laboratories in fulfilling this need. Some of these options are generic systems not targeted toward biomedical research[25], some are commercial systems developed for very specific research markets[26-28], and others are designed for the biomedical research field as a whole, driven by past experiences building systems for research labs[1, 2, 24]. We have explored these various options, and have considered the benefits and limitations of each to help identify a set of requirements for a system we believe would successfully serve the biomedical research community. These requirements have helped to direct us in our approach and design of the CELO system.

2.1 Existing informatics solutions

NeuroSys

NeuroSys is a system developed at the Montana State University Center for Computational Biology (CCB) in response to the recent growth of digital laboratory data[29]. NeuroSys was designed to reduce the complexity of database software such that biomedical researchers, neuroscientists in particular, are able to install, configure and extend data management systems themselves. The NeuroSys developers believe that the inherent complexity of traditional relational databases present a major obstacle for laboratories building data management systems[2]. The NeuroSys approach attempts to reduce this complexity by using a semistructured XML database rather than a traditional relational database. A useful feature of the system is a tool enabling the end users to generate graphical user interface (GUI) screens for data entry and querying by dragging and dropping widgets into a form. The tool empowers the researchers to create customized interfaces that define the XML database structure themselves, without the need for a database or interface programmer. The developers emphasize how the flexibility of the semistructured data facilitates the implementation of interactively constructed and automatically operational GUI screens[2].

NeuroSys is implemented as a Java Web Start application, requiring users to simply install the freely available Java Web Start client software. Because the system is web-based, data can be easily shared over the internet. One major limitation that the NeuroSys developers recognize is the inefficiency of the system due to the semistructured framework[2]. Our review of the system also revealed that the current version of NeuroSys has difficulties handling large data sets, has insufficient querying capabilities, and does not support multimedia data.

SenseLab

SenseLab is a system developed at Yale University as a part of the Human Brain Project. The system was initially developed for integrating various forms of neuronal data, specifically to manage data from experimental research on the olfactory system[30]. These data are collected at the genetic, synaptic, neuronal, brain-pathway, and behavioral levels of the sensory system and are therefore highly heterogeneous. Representations of objects at these multiple levels also may frequently change as scientific knowledge evolves. In consideration of these specific characteristics of olfactory research, SenseLab is based on the Entity-Attribute-Value (EAV) Data Model[24, 31]. In contrast to the traditional relational database, the EAV Data Model represents attributes and attribute values of data objects as data within a single pre-defined database table rather than as columns of separate tables. Changes in representation of data objects therefore do not require modification of the underlying database schema. The developers of SenseLab argue that this static database schema property of the EAV model enables greater flexibility for representing evolving data objects and facilitates generating user interfaces automatically[3, 24, 31, 32]. The EAV model, however, does not support the complex objects and relationships that can be represented using conventional databases. The correlation between a particular Animal Subject and a specific Treatment, for example, can be easily represented in a relational database but not in an EAV database. The SenseLab developers therefore extended the model in order to support classes and relationships, a feature critical for representing biomedical research data. This extended model, called EAV/CR (EAV with classes and relationships), allows the definition of complex data structures called classes, the creation of instances of classes, and the modeling of interclass relationships[32].

The properties of the neuronal data being managed with SenseLab may very well apply to other types of biomedical research data. The SenseLab developers believe the system can therefore be used for other research areas and have demonstrated this by using its framework to build a prototype pharmacogenetics database[32]. A useful feature of the SenseLab system includes a system management console that helps researchers view and design metadata elements and entity relationships. Another valuable feature is SenseLab's ability to automatically generate usable web-based data entry and query screens based on defined metadata[3]. These screens have some configuration settings, however the limited options may not meet some of the specific needs of certain laboratories. For example, users cannot select which widget types (web form input fields) to be displayed in the interface for adding new items into the database. Other limitations of the system include less powerful and less efficient querying capabilties as compared to traditional SQL queries[3, 32]. Our review of the system has also revealed that, similar to NeuroSys, SenseLab does not currently support multimedia data.

Microsoft Access

Microsoft Access[25] is a popular laboratory data management solution. Access is a solid commercial product developed to make database and interface building easy and flexible. Access is a generic system designed to make it less daunting for the novice user to build a relational database management system. It contains multiple utilities, such as wizards and interface design tools, that helps researchers to customize systems themselves. Unlike NeuroSys and SenseLab, Access provides support for multimedia files and has strong querying capabilities based on SQL. Although it is possible to make an Access system run on the web, it is not a simple process to implement, and therefore introduces a challenge for sharing data. Although Access provides users with a variety of tools to build a database system, developing a data management system is still complex and requires a substantial learning curve. Fundamentally, Access at its core remains a relational database management system (RDBMS) and using it effectively requires learning to use and program an RDBMS despite the sophisticated GUIs, tools, and wizards. Because the system is not specifically designed for biomedical research, it also does not consider the needs specific to the research domain, such as special handling of images, a file type commonly used in today's research lab.

Another drawback of the Access option is that the default Access database also cannot accomodate very large data sets. Systems can migrate to Microsoft SQL Server[33] which can support much larger data sets, however, this option is expensive and requires additional time to setup and maintain[34].

WIRM

Web Interfacing Repository Manager (WIRM) is a toolkit that provides informaticists with a set of scripts and utilities facilitating custom code development for a laboratory data management system[1, 35, 36]. We have used the toolkit to build systems such as the Eye Lab image repository[22, 23] and an experiment workflow manager for brain mapping studies[37]. An attractive feature of WIRM is its support for multimedia files. The system has a built-in mechanism for uploading files through a web interface and organizing the files using a combination of filesystem and database utilities. Extensive support for handling images uses pre-existing image modules for actions such as image conversion into web viewable formats and automatically generating thumbnails[20, 36].

Similar to SenseLab, WIRM was also designed to facilitate schema evolution and automatic generation of user interfaces for creating and editing items. As with SenseLab, however, these automatically generated interfaces have limited configuration options and therefore do not always meet the specific needs of a laboratory. In our experiences with WIRM, the automatically generated interfaces were not exactly what were needed by a laboratory, and we often were required to write custom scripts to build custom interfaces. Custom scripts had to be written, for example, to simply display user friendly labels for database tables and table columns in the interfaces for creating or viewing database items.

An informaticist is also needed to install and setup the WIRM system and create the database schema for each laboratory. Researchers must spend valuable research time communicating and clarifying research details in order to ensure that the informaticist can develop a sufficient data management solution.

Our experience working with WIRM has also revealed its limited querying capabilities. Although the system is based on a relational database which allows powerful SQL queries, the WIRM interface limits the types of queries that can be constructed to those that retrieve data from only a single database table. The interface also requires users to have some SQL programming knowledge.

Laboratory Information Management Systems (LIMS)

Many Laboratory Information Management Systems (LIMS) are feature-rich, polished commercial software systems. There are nearly a hundred options that labs can choose from, including vendors such as LabVantage[27], StarLIMS[26], and LabWare[28]. When working with LIMS vendors, research laboratories can create very sophisticated systems with features ranging from inventory and project management to bar coding systems. LIMS tend to be targeted towards larger commercial labs, however, and are therefore very expensive, with “low cost” options starting at a couple thousand dollars, plus personnel resources[38]. A commercial LIMS system is therefore not a plausible option for many smaller academic or non-profit research laboratories. Because LIMS are proprietary products, research labs are also unable to modify the source code of many of these products in order to satisfy unique needs. Add-on customizations must be negotiated with the LIMS vendor, potentially further raising the cost of the informatics solution[38].

Commercial LIMS also tend to target very specific markets, such as the pharmaceutical, environmental, and petrochemical corporations, each of which have relatively standard workflow processes and data types. Many biomedical research laboratories do not fall under one of these standard categories and require a system that enables the definition of much more customized data types. The LIMS option is therefore not an ideal solution for these types of laboratories[1, 39].

There also exist a small number open source LIMS that are freely available. Although these systems do not introduce the cost issues that the commercial systems do, they do suffer from the same limitation that they tend to target specific research operations or domains. For example, caLIMS is an open source system developed by the National Cancer Institute for automating laboratory workflow[40]. The system helps researchers manage projects and inventory using features for handling predefined types such as supplies, samples, assays, and protocols. Flow LIMS is another system developed at the Fox Chase Cancer Center for managing protocols and results specifically for flow cytometry experiments[41]. Gnosis LIMS, on the other hand, is an open source project aimed at creating a customizable system that can be utilized by any laboratory[42]. This system, however, is currently being developed by volunteers, is undergoing design changes, and is not expected to be completed anytime in the near future.

2.2 Requirements for a laboratory data management system

Using our previous experience building laboratory data management systems and the lessons learned through our review of existing informatics solutions, we have devised a list of requirements that we believe are essential for a successful biomedical laboratory data management system. To preface the detailed descriptions of each requirement, we provide a matrix summarizing the review of the existing informatics options based on their ability to satisfy these requirements (Table 1). This matrix demonstrates that the existing systems each have their strengths, but that none of the systems meet a substantial portion of the requirements.

Table 1. Existing System Comparison Matrix

	Inexpensive	Short Development Time	Customizable graphical user interfaces	Features facilitating database design	Support for diverse data types	Powerful querying capabilities	Support for sharing over the internet	Plugins for customizations	Evolution of Data Representation
NeuroSys	X	X	X	X			X		
SenseLab	X	X		X			X		
Microsoft Access		X	X	X	X	X		X	
WIRM	X				X	X	X	X	
Commercial LIMS					X	X	X	X	

Inexpensive

Commercial products, such as LIMS, can often be too expensive for research laboratories, particularly smaller academic or other non-profit labs. Many open source solutions, such as NeuroSys, Senselab and WIRM described earlier, are freely available and additionally allow needed source modifications without additional costs[1, 2, 24]. These open source options, however, may have hardware and personnel requirements that are still too costly for some laboratories to invest in. Ensuring that required hardware is affordable and reducing the need for an informaticist to build a custom system is a priority for many labs in need of an informatics solution[2, 4, 17].

Short Development Time

In our experience, building customized data management systems for research laboratories can take weeks, months or even years. Although not documented in the published literature, our observations from working with these labs indicate that performing biomedical research can be very time sensitive and laboratories sometimes cannot afford to wait long periods of time for a system to be developed. Laboratories we have worked with have often used traditional methods like spreadsheets for recording research data while waiting for an informatics solution to be completed. Once the system was completed, researchers then had to transfer the data, a process that proved to be time consuming and disruptive to the research workflow. Creating working systems quickly not only allows researchers to take advantage of valuable features earlier, but also reduces the hindrance of transferring existing data.

Customizable graphical user interfaces

User interface design greatly affects the usability of a system and therefore also plays a role in system acceptance[43, 44]. Graphical user interfaces, however, can be time consuming to create. As described earlier, some existing systems such as SenseLab and WIRM automatically generate interfaces based on the defined database schema[1, 24]. Although efficiently created, these default interfaces may not exactly suit the specific needs of the end users. The ability to customize particular aspects of the GUI, such as specifying labels and widgets, can help reduce the need to write interfaces from scratch, decreasing system development time[45].

Features to facilitate database design

One of the most challenging aspects of building a data management system is designing a database schema to effectively represent research data. The research scientists themselves best know their data, yet may not have any database design experience. On the other hand, an informaticist may have a good background designing databases, but does not fully understand the research data to be modeled. This leads to a need for close communication between researcher and informaticist and may require a large learning curve on both sides. Features that help facilitate database design can help expedite the development process. Ideally, researchers would be provided with the tools necessary to build their laboratory databases themselves, without being required to have an extensive database background. Several efforts have been made within the informatics field to develop tools to assist users with designing complex database schemas[2, 12, 13, 18]. NeuroSys, for example, implements a drag and drop tool for creating user interfaces, effectively modeling the XML database representing research data[2].

Support for diverse data types

The kinds of data being collected in the biomedical research field are growing rapidly. One of the driving forces of the design of the SenseLab system is the need to manage the heterogeneous types of data collected through experimental studies of the olfactory system. SenseLab allows end users to define their own data types to represent unique kinds of research data[32]. A major limitation of commercial LIMS is their use of predefined industry standard data types that can not flexibly model the diverse data types of other research areas[1].

Many laboratories are also collecting digital file types for research. These file types include multimedia data, such as images[14, 15, 46-49], as well as unique filetypes generated or utilized by domain specific applications[19, 50]. As the advances in and availability of computer technologies increases, and as the costs decrease, the use of these types of digital media in the research lab will grow. Management of multimedia files is therefore essential for a data management system to handle current research needs.

Powerful querying capabilities

As the types of data being collected in the biomedical research lab is growing, so is the amount. One of the major reasons labs are finding a need to invest in informatics solutions to manage data is because of the challenges of managing such large amounts of data[7, 11, 16]. Data management involves not only methods for storing and organizing data, but also methods for finding and retrieving data[3, 4, 8]. Traditional relational databases have well supported querying languages such as SQL that enable powerful queries for finding data. Systems that use non-traditional database models, such as the semistructured database model of NeuroSys and the EAV data model of Senselab, have weaker querying capabilities[2, 3]. A challenge with SQL, however, is providing interfaces that biomedical researchers without SQL knowledge can use and understand.

Support for sharing over the internet

Multi-laboratory research collaborations are becoming more common as many current research efforts require multidisciplinary expertise or are too large scale to be tackled by individual labs. Methods for collaborating laboratories to easily and efficiently share data are therefore important and essential to incorporate into a data management system. Most systems developed specifically to support research collaborations have used web technologies[4, 6, 7, 16, 21]. Utilizing the web is a natural route for sharing data, as the internet is widespread, has multiple well-supported and freely available technologies, and is already utilized for several public biomedical research tools such as the PubMed[51] and Genbank[52] databases. Using the internet as an interface to manage data also supports the potential of someday integrating multiple separate research efforts as well as existing public databases.

Plugins for customizations

Through our experiences building data management systems for research laboratories, we have discovered that each laboratory has specific system feature requests that are likely to not be applicable to other research labs. Such unique customizations cannot be included in a generalized system. The ability to add-on such lab specific features is therefore necessary in order for the system to satisfy each laboratory's unique needs. The concept of plugins is common among software applications as a method to integrate custom functionality into a generic system. For example, plugins for web browsers include integrating multimedia players for viewing multimedia data such as movies and 3-D models. Allowing research laboratories to implement plugins for customizations increases the value of a data management system[16, 53].

Evolution of Data Representation

One reason the developers of SenseLab selected to use the EAV data model is because changes in data representation that require the addition of attributes to a data class does not require an underlying database schema redesign. The SenseLab developers, however, also note that a limitation of their system is the lack of support for changing an attribute's data type after data already exist[32]. The critical system requirement that the developers are addressing when discussing these benefits and limitations of their system is the ability to evolve data representations. Acquisition of new scientific knowledge, modifications to ongoing experiments, and a greater understanding of data collected are only some of the reasons why laboratories might want to modify the way data have been represented in a database. A mechanism for easily and efficiently evolving the data representations while minimizing data loss is a valuable system feature[8, 12, 15].

Chapter 3: CELO System Architecture

The CELO system was designed with a focus on satisfying the nine requirements shaped by our review of existing informatics solutions and through our past experiences building laboratory data management systems. CELO currently satisfies seven of these requirements, more than any of the existing systems in our review (Table 2). The importance of the two requirements that CELO does not currently satisfy, plugins for customizations and evolution of data representation, is emphasized by observations made through our system evaluation that will be discussed later.

Table 2. Existing System and CELO Comparison Matrix

	Inexpensive	Short Development Time	Customizable graphical user interfaces	Features facilitating database design	Support for diverse data types	Powerful querying capabilities	Support for sharing over the internet	Plugins for customizations	Evolution of Data Representation
NeuroSys	X	X	X	X			X		
SenseLab	X	X		X			X		
Microsoft Access		X	X	X	X	X		X	
WIRM	X				X	X	X	X	
Commercial LIMS					X	X	X	X	
CELO	**X**	**X**	**X**	**X**	**X**	**X**	**X**		

The seven requirements that CELO does meet is satisfied by features such as automatic generation of a laboratory data management system framework using a simple web registration form, configuration tools for customizing web user interfaces, a template system for generating pre-defined database schemas, a configurable permissions system for data security, and utilities for storing and organizing multiple file types. These features are captured within five major components that make up the CELO system: CGI scripts and libraries (Figure 3A), the CELO main database (Figure 3B), a collection of XML templates (Figure 3C), a set of laboratory specific MySQL databases (Figure 3D), and a set of laboratory filesystem directories (Figure 3E).

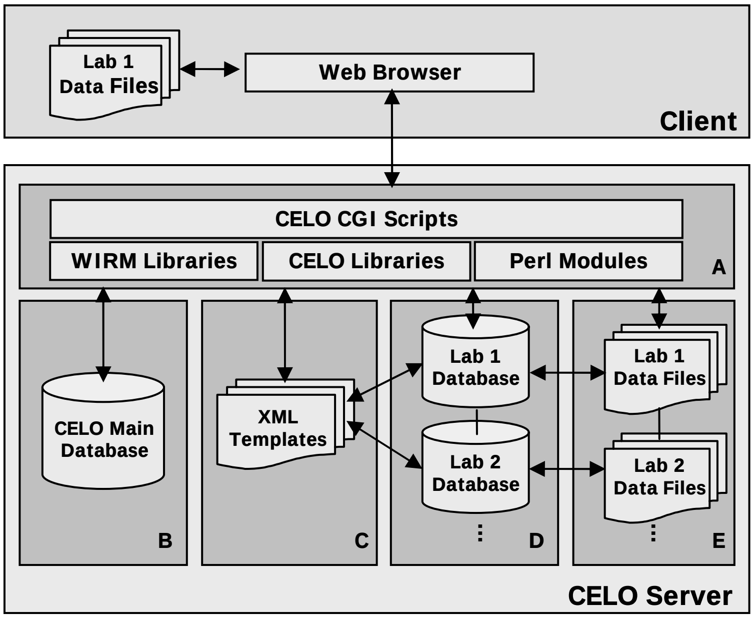
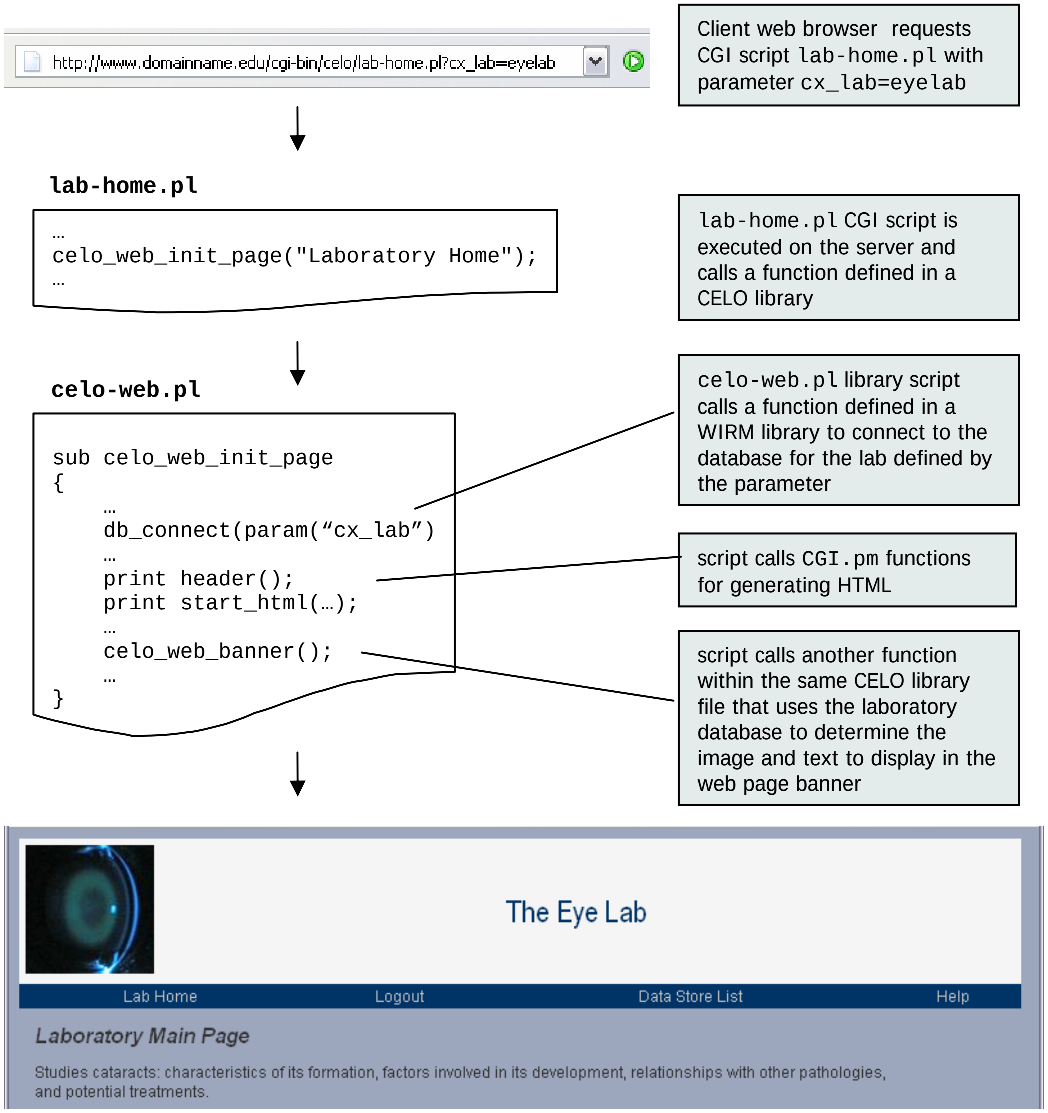


Figure 3. CELO System Architecture

3.1 CGI Scripts and Libraries (Figure 3A)

The CELO Common Gateway Interface[54] (CGI) scripts are computer programs that a client, generally a web browser from the computer of a biomedical research lab, must access in order to use features of the CELO system. Users execute the CGI scripts by simply requesting the appropriate web address, along with any necessary parameters, using their web browser. The CGI scripts utilize functions from three different sets of libraries, the WIRM libraries, the CELO libraries, and the public Perl modules, in order generate the HTML[55] specifying the web interface to be displayed in the client web browser (Figure 4).



http://www.domainname.edu/cgi-bin/celo/lab-home.pl?cx_lab=eyelab

Client web browser requests CGI script `lab-home.pl` with parameter `cx_lab=eyelab`

`lab-home.pl`

```
…
celo_web_init_page("Laboratory Home");
…
```

`lab-home.pl` CGI script is executed on the server and calls a function defined in a CELO library

`celo-web.pl`

```
sub celo_web_init_page
{
    …
    db_connect(param(“cx_lab”)
    …
    print header();
    print start_html(…);
    …
    celo_web_banner();
    …
}
```

`celo-web.pl` library script calls a function defined in a WIRM library to connect to the database for the lab defined by the parameter

script calls `CGI.pm` functions for generating HTML

script calls another function within the same CELO library file that uses the laboratory database to determine the image and text to display in the web page banner

The Eye Lab

Laboratory Main Page

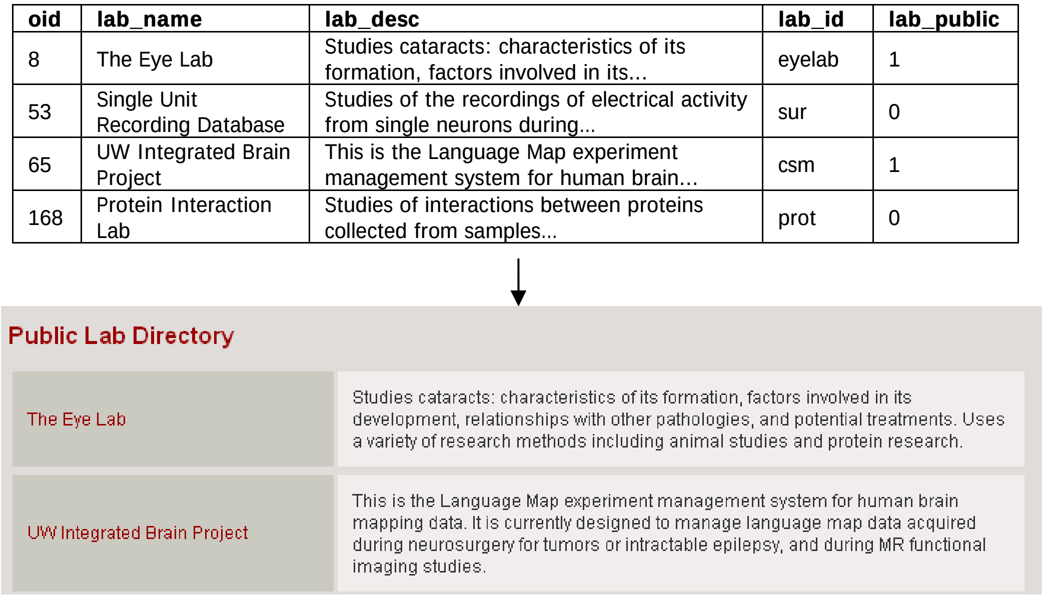
Studies cataracts: characteristics of its formation, factors involved in its development, relationships with other pathologies, and potential treatments.

Figure 4. Interaction between CGI scripts and libraries

The CELO CGI scripts and libraries are implemented in the Perl programming language[56] and reside on the CELO server. A set of Perl libraries developed for the WIRM toolkit[1, 36] described earlier was used as a base for this component of the system. We selected WIRM as a base for CELO because it is open source software that provides an excellent framework for a web-based system, as well as utilities for handling multimedia files. Open source Perl modules that are freely available[57], such as CGI.pm for generating HTML[58] and SimpleObject.pm for parsing XML[59], were also utilized. We developed additional CELO libraries to perform tasks specific to the CELO system, such as generating web page content customized for each laboratory, defining laboratory database structures, generating and parsing web forms to create, edit or view database items, and logging system usage messages.

3.2 CELO Main Database (Figure 3B)

The CELO Main Database stores information useful for every laboratory in the research organization that a given CELO installation serves. Each laboratory can access a web page to view data stored in the CELO Main Database, such as a directory of laboratory systems and details about available templates that any lab can use for building a data management system. There is only one CELO Main Database per CELO installation and it is created on the server when the CELO installation script is executed. The CELO Main Database is a MySQL database[60] consisting of two tables: the `Research_Labs` table and the `Templates` table. The `Research_Labs` table stores general information about each laboratory system including a name, a lab id, description, and whether to list it in the CELO lab directory that is accessible from the CELO Home Page. This table is empty upon CELO installation and is populated with a new entry when a laboratory submits the web registration form. Entries with the `lab_public` column value set to 1 are included in the lab directory which displays each public laboratory's name, description and a link to the Lab Home Page (Figure 5).



oid	lab_name	lab_desc	lab_id	lab_public
8	The Eye Lab	Studies cataracts: characteristics of its formation, factors involved in its...	eyelab	1
53	Single Unit Recording Database	Studies of the recordings of electrical activity from single neurons during...	sur	0
65	UW Integrated Brain Project	This is the Language Map experiment management system for human brain...	csm	1
168	Protein Interaction Lab	Studies of interactions between proteins collected from samples...	prot	0

Public Lab Directory

The Eye Lab	Studies cataracts: characteristics of its formation, factors involved in its development, relationships with other pathologies, and potential treatments. Uses a variety of research methods including animal studies and protein research.
UW Integrated Brain Project	This is the Language Map experiment management system for human brain mapping data. It is currently designed to manage language map data acquired during neurosurgery for tumors or intractable epilepsy, and during MR functional imaging studies.

Figure 5. Relationship between the CELO Main Database and the Public Lab Directory

The `Templates` table stores information about each defined CELO template file. Details about the contents and use of these templates will be described later. Upon installation of CELO, entries for each of the default templates are entered in the Templates table. We created these default templates using our past experience building data management systems for real biomedical research laboratories. The templates can immediately be utilized by laboratories that have registered for a CELO-generated system and can also be used for demos or as examples for building additional templates. The `Templates` table stores details about the available templates including a name, description, an author ID, the authoring organization, date created, and the filepath to the template file in the server filesystem. The table is populated with additional entries as laboratories save new templates.

3.3 Laboratory Databases (Figure 3D)

While the CELO Main Database stores generic data relevant for all research labs in an organization, the Laboratory Databases store data specific to each lab. The Laboratory Databases store the research data, as well as customization settings, for each laboratory system. A dedicated MySQL database[60] is created for each laboratory that submits a web registration form, and so each CELO installation can contain multiple laboratory databases. Each database contains a set of default tables that play key roles in features such as a permissions system, a mechanism for customizing how to represent, organize, and view research data, a system usage log, and predefined representations of data types commonly used in the laboratory. We describe here how each of the default database tables contributes to the implementation of these features.

The `User_Group`, `User`, and `Session` tables are important components of the CELO permissions system (Figure 6A). The permissions system was developed to help laboratories that need to share data control who has access to particular features of the system. Labs can create custom User Groups which are assigned a set of access permissions to particular laboratory system features, such as viewing or adding items. Users are then created and assigned to a specific User Group. CGI scripts are provided for Users to log in to the laboratory system which then generates a Session and gives them access to system features specified by their User Group. Sessions manage the CELO log in state such that permissions are propagated as users navigate between different pages of the system. Sessions are deleted once a user logs out.

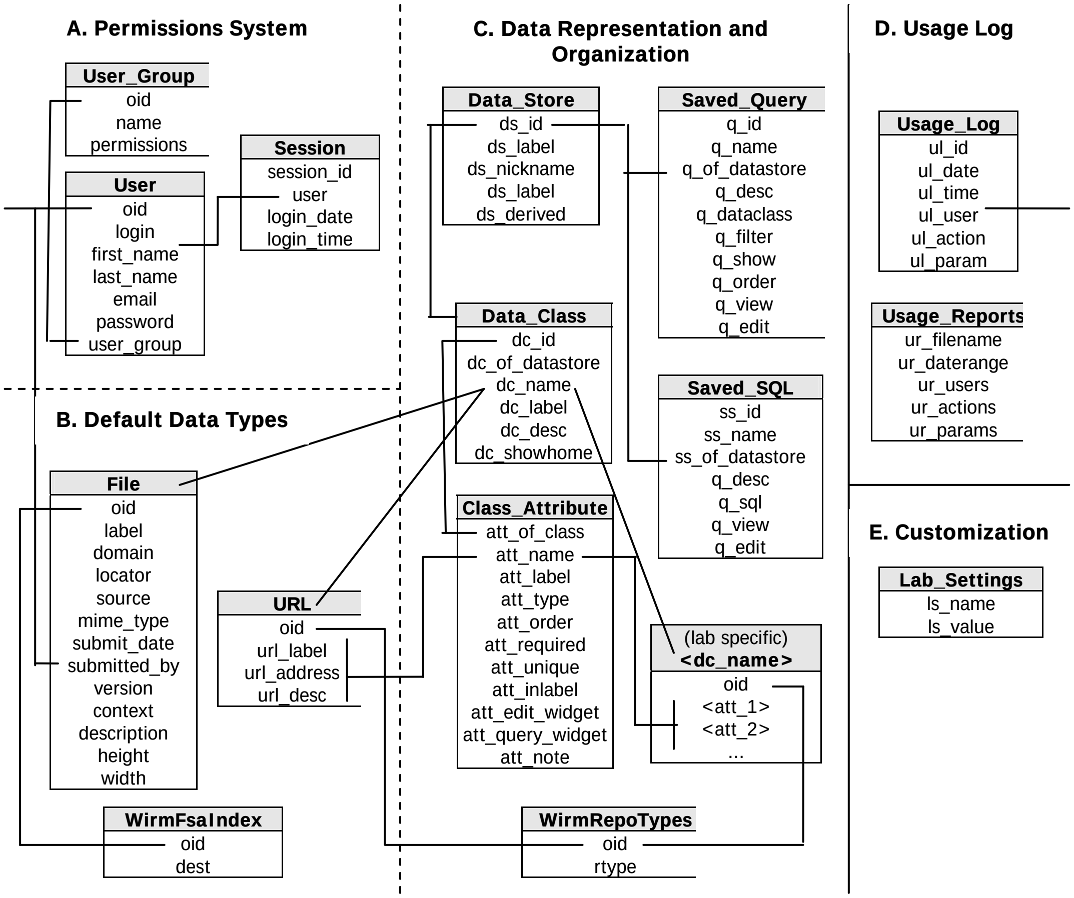


Figure 6. Laboratory Database Tables

Two types of data that are growing in use in the research lab are computerized files, such as digital images, and web addresses (URLs), such as links to an entry in a public database such as GenBank[52]. We therefore defined two data types to represent files and URLs that are automatically available for use in any CELO-generated laboratory system (Figure 6B). Instances of a file are stored in the `File` table which specifies a variety of information including, among other details, the source of the uploaded file, the location of the stored file on the server, and the file type. The details stored in the `File` table play a role in how CELO handles the organization and visualization of data files. The actual files uploaded to the system are stored in the laboratory filesystem directory component of CELO that is described later. Similar to the file data type, instances of a URL are stored in the URL table and the system performs special processing for displaying URL items as web links.

File and URL are generic data types that research laboratories are likely to need, however, most labs will also want to define customized data types for representing research data. Labs are also likely to want to configure the user interfaces for creating, querying or viewing instances of these data types. The combination of the data type definitions and their corresponding interface settings are termed *Data Classes* in the CELO system. Data Classes are a major component of the *Data Store*, a construct developed to help researchers organize data by grouping related data together (Figure 6C). For example, data from separate experiments may be organized into separate Data Stores. A generic CGI script generates a web portal for accessing data belonging to a given Data Store. Data Stores are defined by a set of Data Classes and Saved Queries. A Data Class corresponds to a laboratory-unique database table for storing research data and entries in the Data_Classes and Class_Attributes tables for specifying additional details. These details include a brief description of the Data Class, user friendly labels for displaying table column names, and specifications of the widgets to provide users for adding or querying for items. Queries composed to retrieve research data are saved in the Saved_Query and Saved_SQL tables. The two different tables specify queries composed using two different methods. We call the queries that are saved in the Saved_Query table *simple queries* because users generate them using a simple query form that is automatically generated using metadata for a given Data Class. Simple queries are easy and fast to create, yet are not very flexible. For example, simple queries can only retrieve data from a single database table at one time. The queries that are saved in the `Saved_SQL` table, on the other hand, allow users to construct more complex queries such as those joining data from multiple database tables. These *advanced queries* are more difficult and time consuming to create and require knowledge of SQL commands and syntax, but are much more powerful. More details about how end users create and use both the simple and advanced queries will be presented in the system workflow chapter.

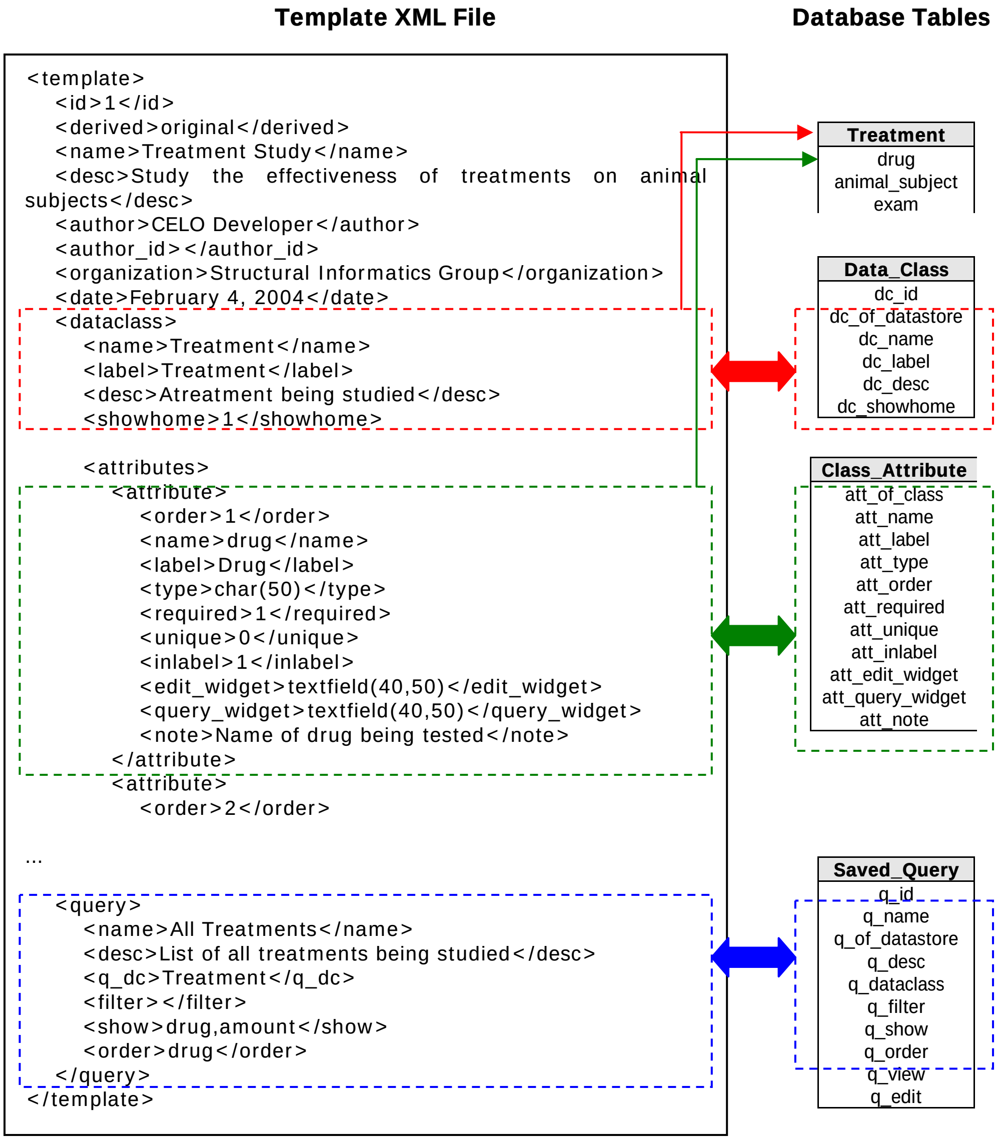
In order to help labs keep track of activity within a laboratory data management system, entries are added to the `Usage_Log` table each time particular actions are performed (Figure 6D). Each entry contains details about the action performed such as the date and time, the user who performed the action, what the action was, and any additional parameters describing the action. CELO provides features for querying the Usage Log and saving the query results into an HTML report. Details about these saved reports, including the query parameters and the location of the saved HTML file, are entered in the `Usage_Reports` table. The information in the table is used to generate an interface for users to view and retrieve the saved reports.

Several laboratory system customizations are stored in the `Lab_Settings` table (Figure 6E). CELO defines a set of lab settings that are used to determine some aspects of the web page user interface that are specific to each lab. For example, values for the settings for lab name, description and location determine the text to display in a laboratory’s web page banner and Laboratory Home Page. Other settings include which links to include in the navigation bar, what information to include on the Laboratory Home Page, and color, size and spacing properties for displaying tables in the web pages.

3.4 XML Templates (Figure 3C)

The CELO template system was developed to assist researchers with designing database schemas for representing research data. Each template predefines data representations that labs can use for very specific research tasks. A lab, for example, can select a template to automatically generate the database representation for a chemicals inventory or for an experimental study of the affects of various treatments on the development of a disease. The aim of the template system is to help researchers without a database background define data representations and to foster database schema reuse and sharing, helping to make schema design more efficient. Each template consists of template metadata, including a template name, description, author and date of creation, as well as definitions for a set of Data Classes and Saved Queries.

The templates are written in XML (Extensible Markup Language)[61], a markup language used to describe data in a structured format and which is becoming a standard among software development. CELO CGI scripts provide interfaces for users to browse details about existing templates and to select a template to populate a newly created Data Store with a default set of Data Classes and Saved Queries. A lab performing a study of the effects of creatine on cataract development in mice might, for example, select the Treatment Study template that defines Data Classes for *Animal Subject*, *Treatment*, and *Exam.* The template XML elements are parsed and transformed into entries of the appropriate laboratory database tables. A new table is also generated for each defined Data Classes. A CGI script is also available to perform the reverse transformation using the database structure of an existing Data Store to construct an XML template. The transformations between an XML template and CELO laboratory database tables are illustrated in Figure 7.



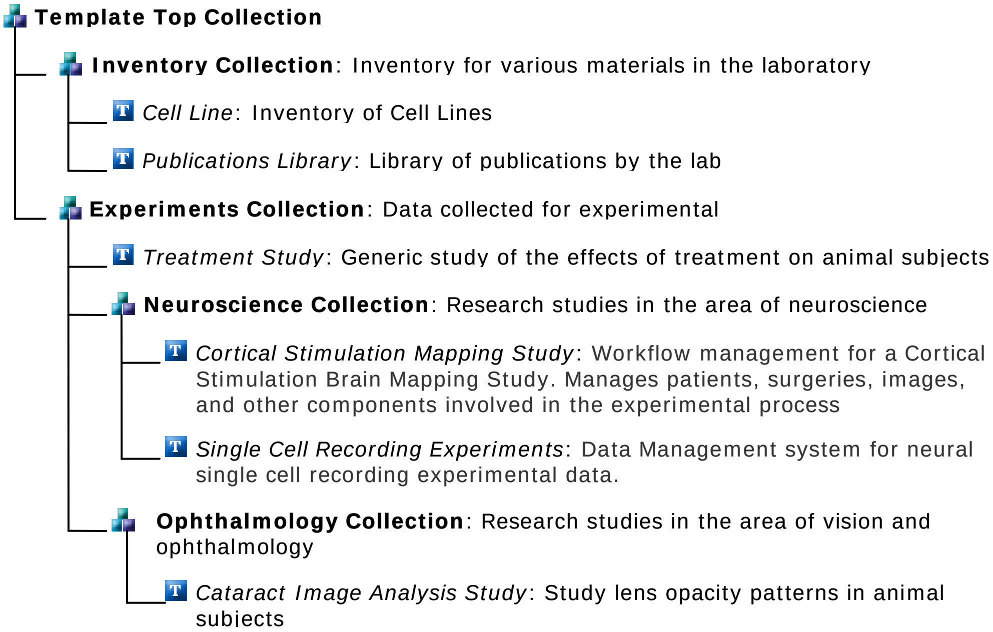
```
<template>
  <id>1</id>
  <derived>original</derived>
  <name>Treatment Study</name>
  <desc>Study the effectiveness of treatments on animal
subjects</desc>
  <author>CELO Developer</author>
  <author_id></author_id>
  <organization>Structural Informatics Group</organization>
  <date>February 4, 2004</date>
  <dataclass>
    <name>Treatment</name>
    <label>Treatment</label>
    <desc>Atreatment being studied</desc>
    <showhome>1</showhome>

    <attributes>
      <attribute>
        <order>1</order>
        <name>drug</name>
        <label>Drug</label>
        <type>char(50)</type>
        <required>1</required>
        <unique>0</unique>
        <inlabel>1</inlabel>
        <edit_widget>textfield(40,50)</edit_widget>
        <query_widget>textfield(40,50)</query_widget>
        <note>Name of drug being tested</note>
      </attribute>
      <attribute>
        <order>2</order>

...
  <query>
    <name>All Treatments</name>
    <desc>List of all treatments being studied</desc>
    <q_dc>Treatment</q_dc>
    <filter></filter>
    <show>drug,amount</show>
    <order>drug</order>
  </query>
</template>
```

Figure 7. Transformation between XML Template and Database Tables

The template files are organized in a hierarchical manner using a directory structure on the server filesystem. A designated data file exists in each directory that describes the collection of templates that reside in that directory. The CELO installation contains a set of default templates that are readily available for laboratories to browse and use (Figure 8). The default “Inventory” collection of templates is composed of generic templates for tracking cell lines or for creating a library of publications relevant for a lab’s research. These templates were created to demonstrate the range of functions that the system can be used for. A second default template collection is called “Experiments” and contains two sub-collections for research areas with which we have previously worked with. These sub-collections, “Ophthalmology” and “Neuroscience”, contain templates that we created using our experience developing systems for labs and are utilized in our system evaluation described later.



- **Template Top Collection**
  - **Inventory Collection**: Inventory for various materials in the laboratory
    - *Cell Line*: Inventory of Cell Lines
    - *Publications Library*: Library of publications by the lab
  - **Experiments Collection**: Data collected for experimental
    - *Treatment Study*: Generic study of the effects of treatment on animal subjects
    - **Neuroscience Collection**: Research studies in the area of neuroscience
      - *Cortical Stimulation Mapping Study*: Workflow management for a Cortical Stimulation Brain Mapping Study. Manages patients, surgeries, images, and other components involved in the experimental process
      - *Single Cell Recording Experiments*: Data Management system for neural single cell recording experimental data.
    - **Ophthalmology Collection**: Research studies in the area of vision and ophthalmology
      - *Cataract Image Analysis Study*: Study lens opacity patterns in animal subjects

Figure 8. Default Templates and Template Collection Hierarchy

3.5 Laboratory Filesystem Directories (Figure 3E)

In addition to the laboratory database created for each laboratory that registers for a CELO system, a laboratory directory is also created on the server for each lab. This directory stores a lab's research data files which can include a variety of file types such as images, spreadsheets, and text documents. These data files are uploaded to the CELO server by the end users through a web browser. The system automatically stores and organizes the uploaded files and associates them with an appropriate database entry in the `Files` table. The database table entry specifies file metadata such as its filesystem location, the original source path on the client's computer, a file label, and file type. These metadata help determine how to display a file item that a users requests. Image files, for example, are displayed as images embedded in the web page, whereas a spreadsheet file is displayed as a link that launches a spreadsheet application such as Microsoft Excel as determined by the web browser. In addition to data files, lab directories also contain configuration files for web display settings. These files include a cascading stylesheet (CSS)[62] which specifies web page display settings including color schemes and font styles, as well as a lab logo image file to display in the banner of each webpage. The root of the directory tree created for each laboratory is located under the `labs` directory of the CELO installation and is named after the lab ID specified during registration.

Chapter 4: CELO Usage Workflow and System Features

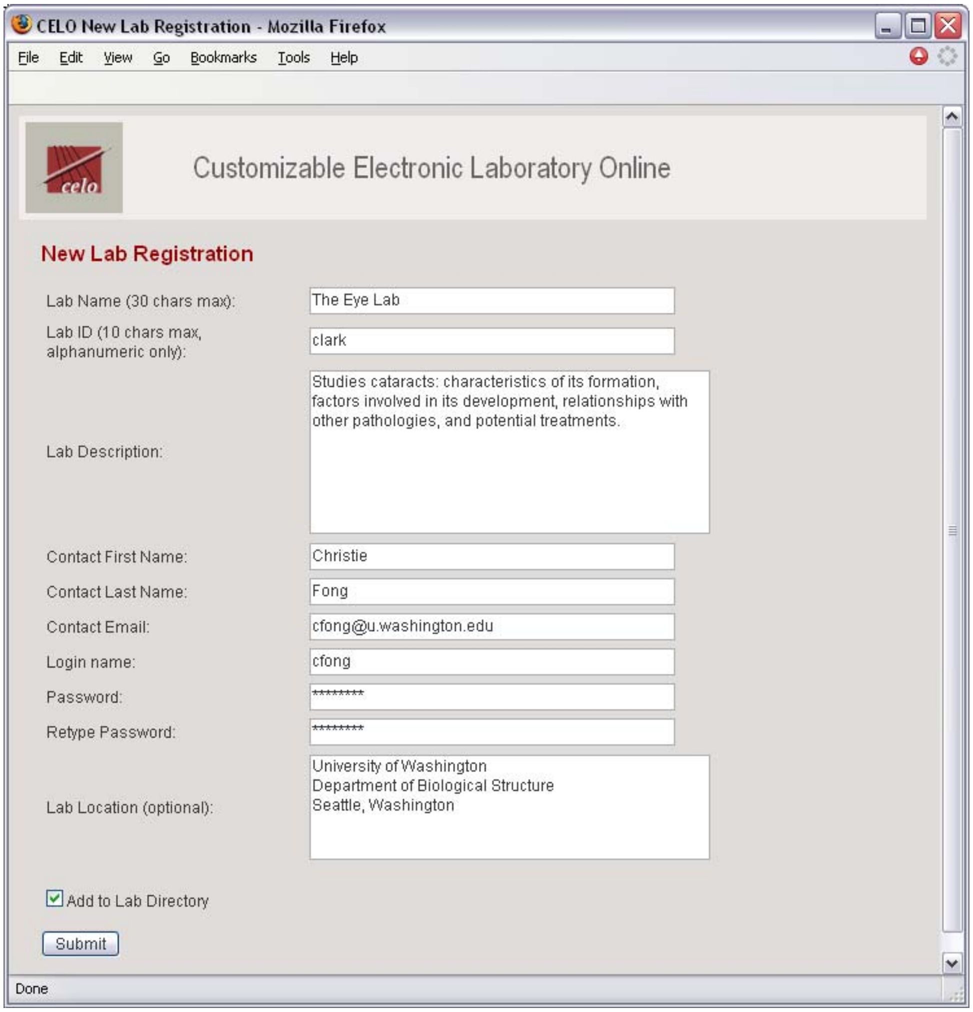
Once CELO installation and setup is complete, any lab with access to the server can register for a new system. Labs will typically perform a set of steps to create, customize, and then use their system (Table 3). We provide a description of each of these steps, which demonstrates many of the core features of the CELO system.

Table 3. Typical System Workflow Steps

1	Register for a New System
2	Log In
3	Create User Groups and Users
4	Create a Data Store
5	Browse for and Use a Template
6	Modify Data Classes
7	Create a New Data Class
8	Enter Items
9	Query for Items
10	Save a Query
11	Save a Template
12	View the Usage Log
13	Customizing Web Display Settings
14	Log Out

4.1 Register for a New System

Many informatics solutions require researchers to install and setup the necessary software components for a new system before it can be used and customized for specific laboratory needs. This step can require a large learning curve and be quite time consuming for someone without previous system administration experience. With the CELO system, this time intensive installation and setup process must be performed only once, and multiple labs can immediately begin using its tools, without having to invest time in the same process. In order to begin using the tools, a lab must merely complete a registration web form. The form contains input fields requesting information such as the lab name, description, unique identifier, and information about a designated contact person (Figure 9).


CELO New Lab Registration - Mozilla Firefox
File
Edit
View
Go
Bookmarks
Tools
Help
celo
Customizable Electronic Laboratory Online
Submit
Done

New Lab Registration

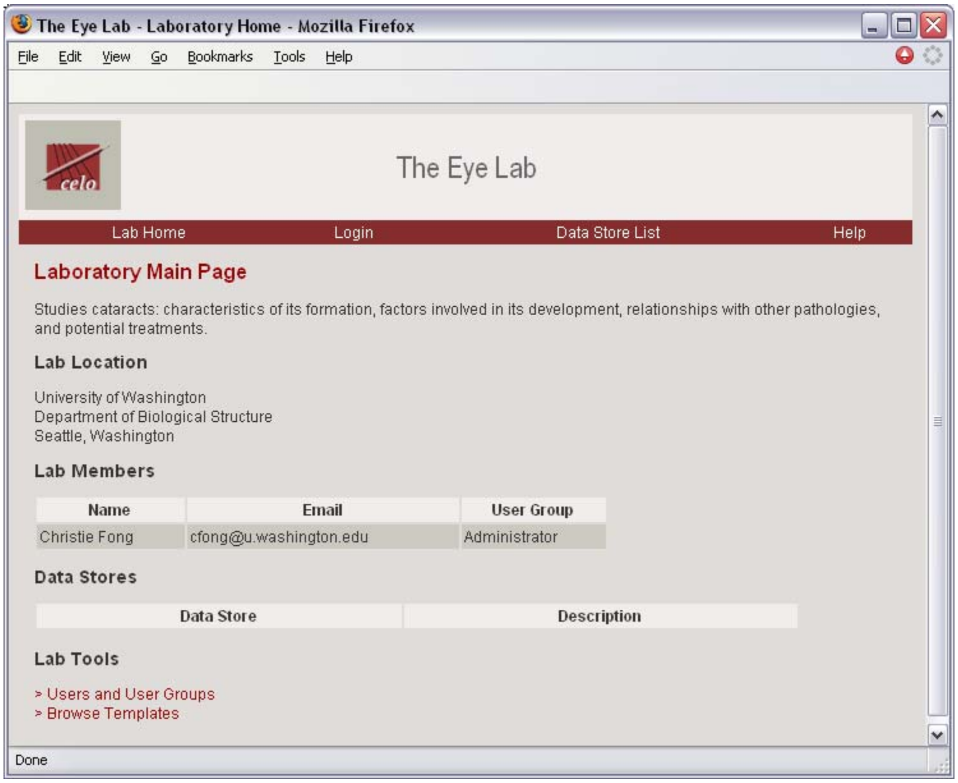
Lab Name (30 chars max):	The Eye Lab
Lab ID (10 chars max, alphanumeric only):	clark
Lab Description:	Studies cataracts: characteristics of its formation, factors involved in its development, relationships with other pathologies, and potential treatments.
Contact First Name:	Christie
Contact Last Name:	Fong
Contact Email:	cfong@u.washington.edu
Login name:	cfong
Password:	********
Retype Password:	********
Lab Location (optional):	University of Washington Department of Biological Structure Seattle, Washington

☑ Add to Lab Directory

Figure 9. Laboratory Registration Page

When the web form is submitted, a new laboratory database and filesystem directory are automatically created on the CELO server. Particular laboratory system settings stored in the database, such as the laboratory name and description, default to values specified in the registration form. Each laboratory's customized web page can then be accessed by requesting

CELO CGI scripts with the cx_lab parameter set to the laboratory's id. The cx_lab parameter indicates to CELO to connect to the appropriate lab database. Settings stored in the database dictate the content and appearance of each dynamically generated web page (Figure 10).



The Eye Lab

Laboratory Main Page

Studies cataracts: characteristics of its formation, factors involved in its development, relationships with other pathologies, and potential treatments.

Lab Location

University of Washington
Department of Biological Structure
Seattle, Washington

Lab Members

Name	Email	User Group
Christie Fong	cfong@u.washington.edu	Administrator

Data Stores

Data Store	Description

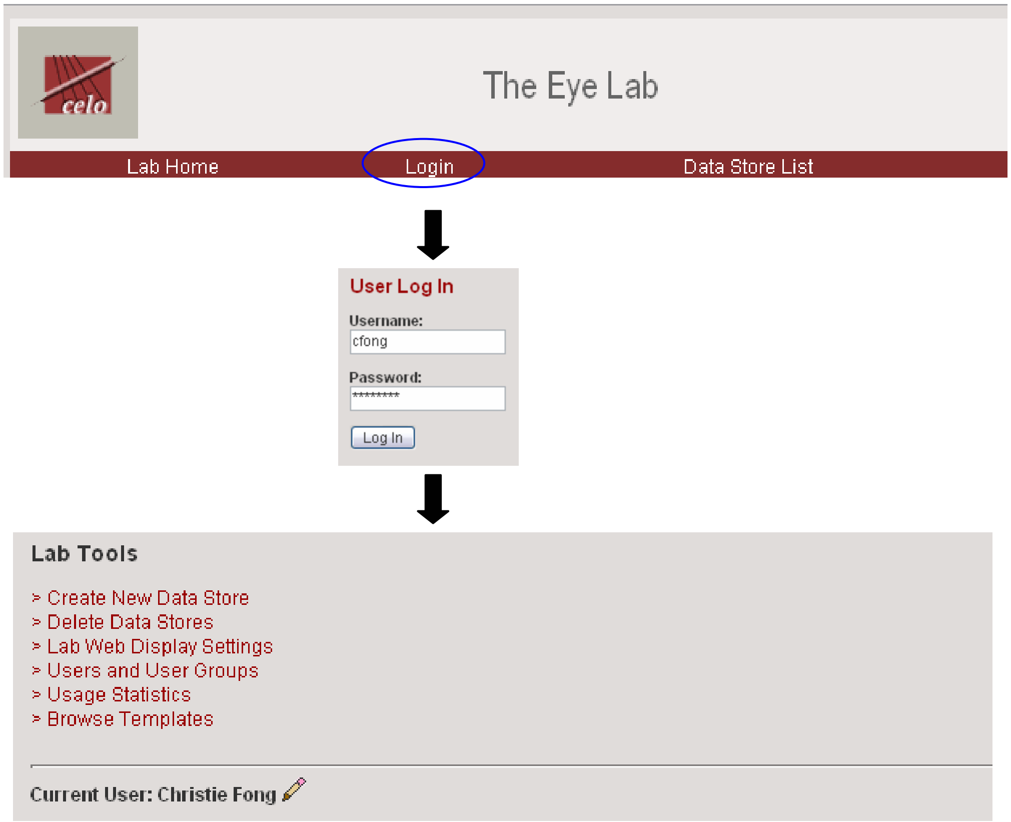
Lab Tools

› Users and User Groups
› Browse Templates

Figure 10. Laboratory Main Page

4.2 Log In

CELO implements a permissions system in order to allow laboratories to control who has access to particular features of their system. Most labs, for example, will not want to allow every user to be able to change the lab system's configuration settings. User Groups and Users are two key components of the CELO permissions system. When a laboratory successfully registers for a system, two default User Groups are automatically created: the *Administrator* User Group and the *Public* User Group. The Administrator User Group has system wide permissions by default. The contact person specified in the registration page is automatically created as a user in the Administrator User Group. This user, therefore, can log in to the laboratory system by selecting the *Log In* link in the lab web site's navigation bar and entering the user name and password specified during registration. Once the user is logged in to the system, the user's name will appear at the foot of each web page, and the *Log In* link will switch to *Log Out*. This user with Administrator privileges can then access a variety of tools for performing actions such as viewing and editing items in the database, as well as configuring the system for the laboratory's custom needs (Figure 11). The Public User Group defines permissions for any user who is not logged in to the system. A Public user by default has permissions to view items in the laboratory database, but cannot edit items nor configure the system. An administrator can easily modify the default permissions for these two User Groups that are automatically generated by CELO.



Username: cfong

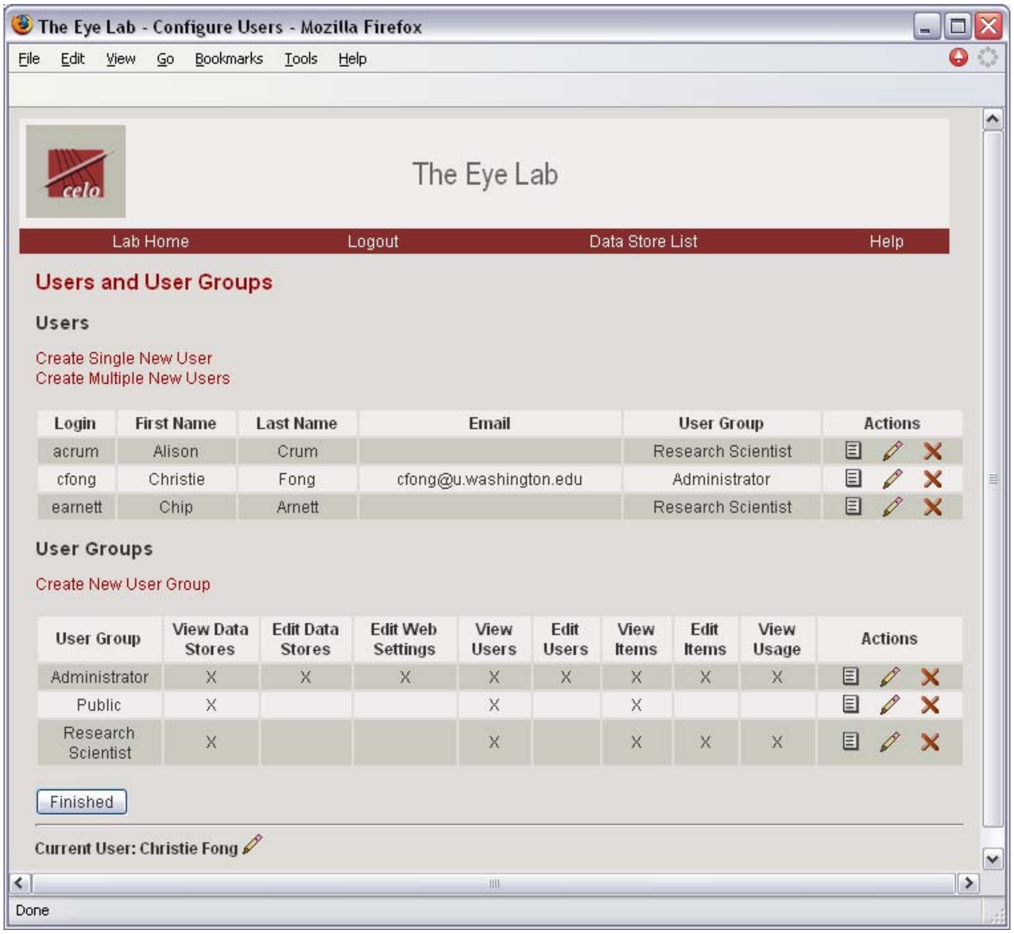
Password: ********

Current User: Christie Fong

Figure 11. User Log In

4.3 Create User Groups and Users

One of the features a logged in Administrator has access to is a tool to create customized User Groups and Users. User Groups are defined by a name, such as “Research Scientist”, and a set of permissions, such as “View Items” and “Edit Items”. Users are created, assigned to a particular User Group and, when logged in, have access to the features defined by their User Group (Figure 12). This customizable permissions system helps laboratories control who has access to certain system features, serving as a valuable security feature and facilitating custom needs for sharing data within a research collaboration.


The Eye Lab - Configure Users - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
celo
The Eye Lab
Lab Home
Logout
Data Store List
Help
Finished
Done

Users and User Groups

Users

Create Single New User
Create Multiple New Users

Login	First Name	Last Name	Email	User Group	Actions
acrum	Alison	Crum		Research Scientist	▤ ✎ ✕
cfong	Christie	Fong	cfong@u.washington.edu	Administrator	▤ ✎ ✕
earnett	Chip	Arnett		Research Scientist	▤ ✎ ✕

User Groups

Create New User Group

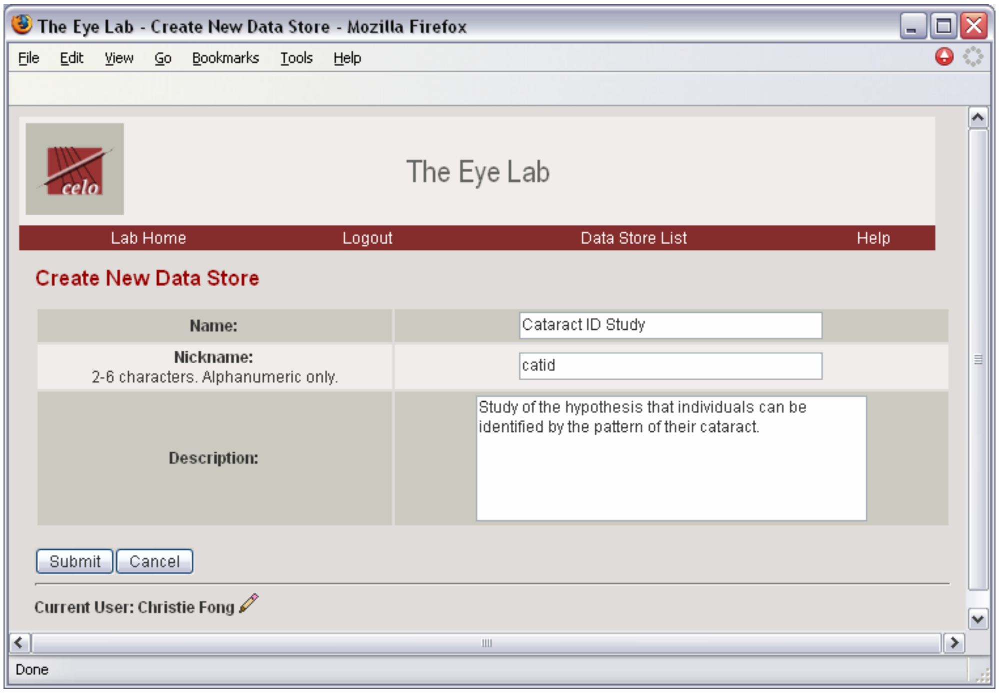
User Group	View Data Stores	Edit Data Stores	Edit Web Settings	View Users	Edit Users	View Items	Edit Items	View Usage	Actions
Administrator	X	X	X	X	X	X	X	X	▤ ✎ ✕
Public	X			X		X			▤ ✎ ✕
Research Scientist	X			X		X	X	X	▤ ✎ ✕

Current User: Christie Fong ✎

Figure 12. Create Users and User Groups

4.4 Create a Data Store

Data Stores are a concept created for the CELO system to help labs keep related research together without being cluttered by other non-related data also stored in the laboratory database. For example, a lab might want to create separate Data Stores to manage data collected from different experiments. Any User with the appropriate permissions can create Data Stores for organizing research data. Data associated with each Data Store can be accessed and configured through a dedicated web page. Users with the appropriate permissions will be able to use a link on the Laboratory Main Page in order to create a new Data Store. The user will be prompted to enter a Data Store name, nickname and description (Figure 13). The nickname is a 2-6 character alphanumeric string that is used as a database table prefix to help differentiate the tables created for each Data Store.



Create New Data Store

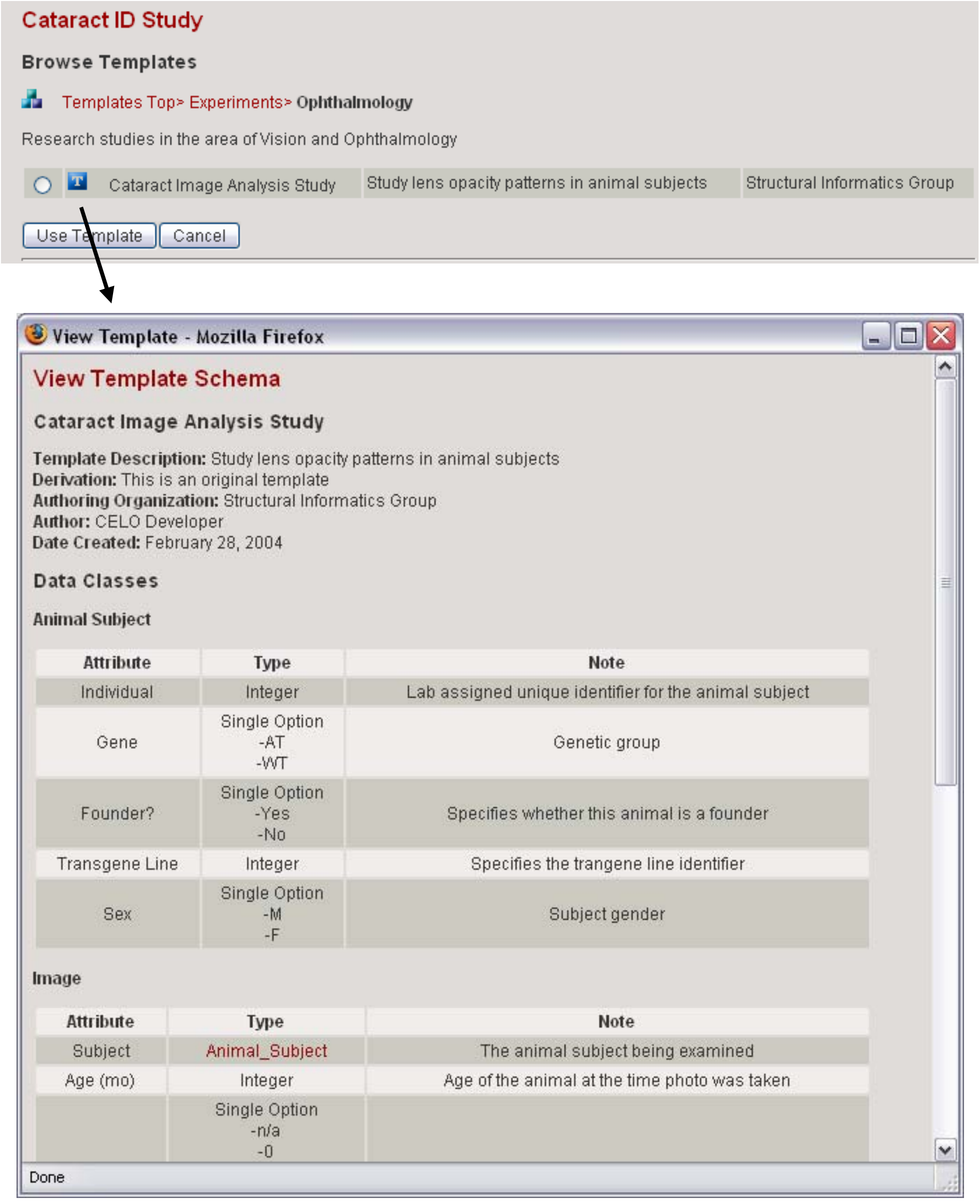
Name:	Cataract ID Study
Nickname: 2-6 characters. Alphanumeric only.	catid
Description:	Study of the hypothesis that individuals can be identified by the pattern of their cataract.

Current User: Christie Fong

Figure 13. Create New Data Store

4.5 Browse for and Use a Template

In order for researchers to start entering data into their database system, they first must define how their research data is represented in the database. CELO uses the concept of a Data Class for defining these representations. An experiment studying the effects of various treatments for a particular disease, for example, might be represented by Data Classes for *Animal Subjects*, *Treatments*, and *Exams*. Once a Data Store has been created, the user will be given an option to define a new Data Class from scratch or to use an existing template to define the Data Classes that make up the Data Store. If a template exists that well describes the type of data that need to be stored, such as data collected through a specific type of experiment, using a template is the easier and more efficient method for creating Data Classes. The available templates can be browsed by navigating through the hierarchical tree of template collections. To assist users with selecting an appropriate template, users can view template details such as the name, description, author, authoring organization, date created, and a list of Data Classes and Queries defined by the template (Figure 14). Once a template is selected and used, the template defined Data Classes and Queries are automatically generated and associated with the newly created Data Store.



Cataract ID Study

Browse Templates

Templates Top> Experiments> Ophthalmology

Research studies in the area of Vision and Ophthalmology

○	Cataract Image Analysis Study	Study lens opacity patterns in animal subjects	Structural Informatics Group

View Template - Mozilla Firefox

View Template Schema

Cataract Image Analysis Study

Template Description: Study lens opacity patterns in animal subjects
Derivation: This is an original template
Authoring Organization: Structural Informatics Group
Author: CELO Developer
Date Created: February 28, 2004

Data Classes

Animal Subject

Attribute	Type	Note
Individual	Integer	Lab assigned unique identifier for the animal subject
Gene	Single Option -AT -WT	Genetic group
Founder?	Single Option -Yes -No	Specifies whether this animal is a founder
Transgene Line	Integer	Specifies the trangene line identifier
Sex	Single Option -M -F	Subject gender

Image

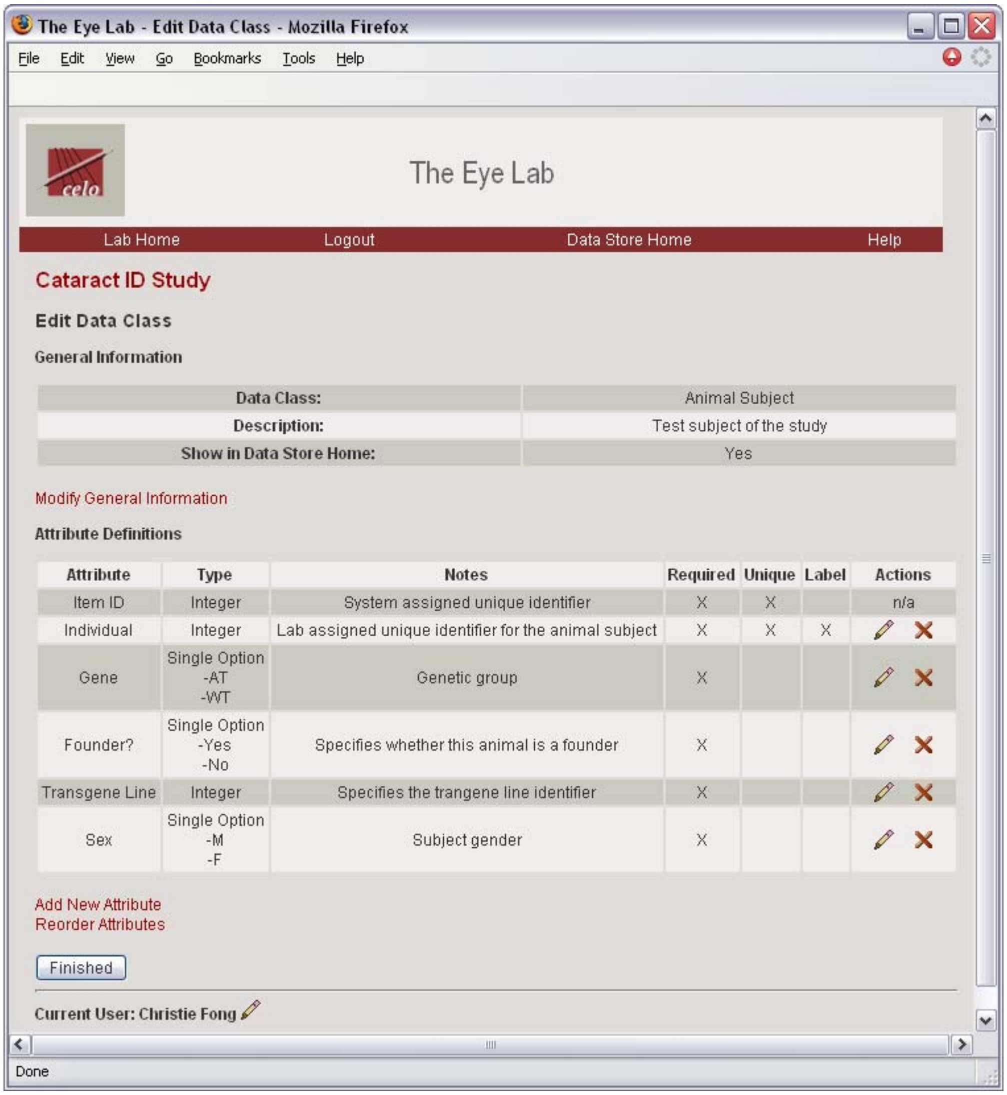
Attribute	Type	Note
Subject	Animal_Subject	The animal subject being examined
Age (mo)	Integer	Age of the animal at the time photo was taken
	Single Option -n/a -0	

Done

Figure 14. Browse Templates

4.6 Modify Data Classes

Templates can help researchers create a database schema for representing certain research data, but they may not exactly fit the specific needs of the laboratory. A template might define an *Animal Subject* Data Class, for example, with attributes for date of birth and sex, but not including an attribute for weight, which might be an important characteristic to record for a specific experiment. Researchers can therefore easily modify the *Animal Subject* Data Class generated by a template to include a new attribute for weight. Configuration tools additionally allow researchers to modify the attribute properties associated with any Data Class. Attribute properties include the label, description, data type, widgets to use for adding, editing or querying, and flags indicating if an attribute value is required, must be unique, or should be included in the label for an item (Figure 15).


The Eye Lab - Edit Data Class - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
celo
Lab Home
Logout
Data Store Home
Help
Done

The Eye Lab

Cataract ID Study

Edit Data Class

General Information

Data Class:	Animal Subject
Description:	Test subject of the study
Show in Data Store Home:	Yes

Modify General Information

Attribute Definitions

Attribute	Type	Notes	Required	Unique	Label	Actions
Item ID	Integer	System assigned unique identifier	X	X		n/a
Individual	Integer	Lab assigned unique identifier for the animal subject	X	X	X	✎ ✗
Gene	Single Option -AT -WT	Genetic group	X			✎ ✗
Founder?	Single Option -Yes -No	Specifies whether this animal is a founder	X			✎ ✗
Transgene Line	Integer	Specifies the trangene line identifier	X			✎ ✗
Sex	Single Option -M -F	Subject gender	X			✎ ✗

Add New Attribute
Reorder Attributes

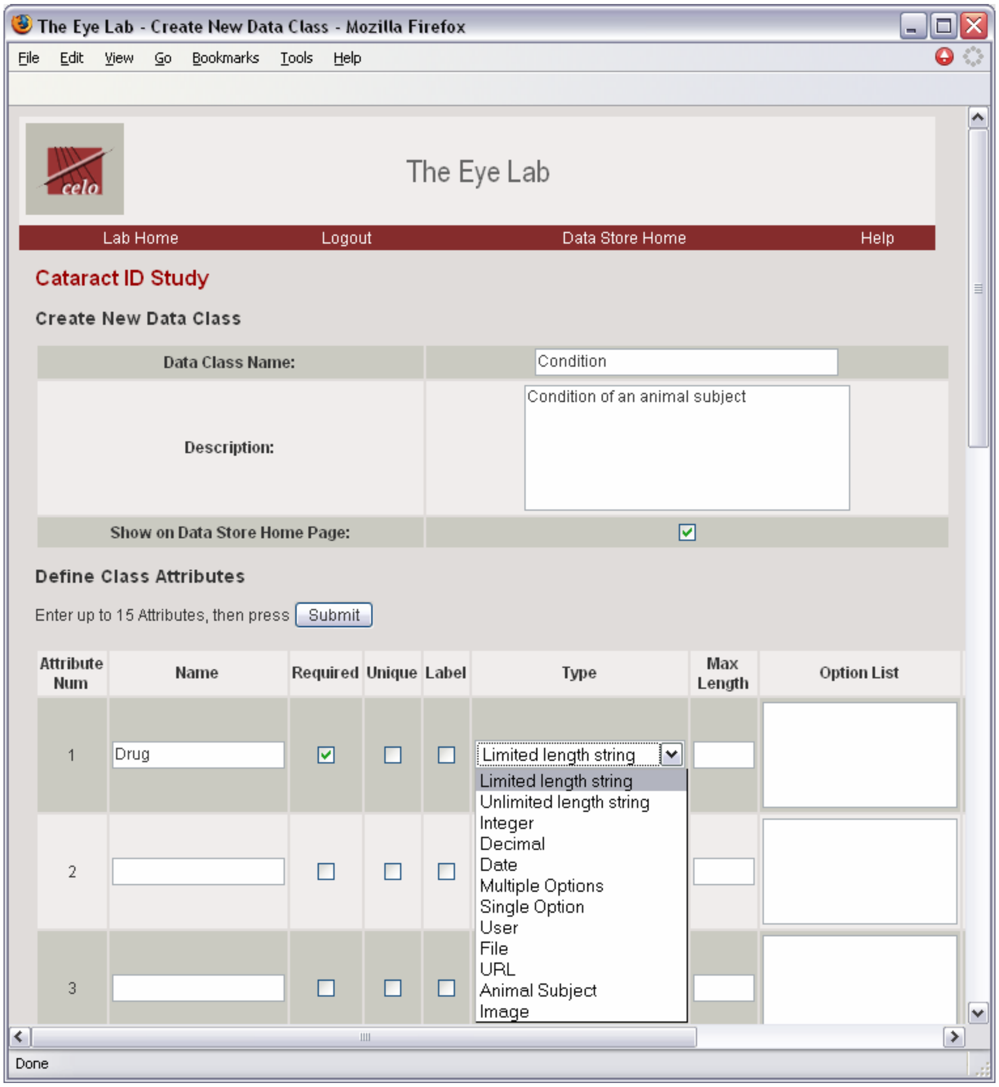
Finished

Current User: Christie Fong ✎

Figure 15. Modify Data Class

4.7 Create a new Data Class

Even after a template has been used to generate a pre-defined set of Data Classes, new Data Classes can still be added. A lab may want to do this, for example, if a template specifies definitions for *Animal Subject*, *Treatment,* and *Exam* Data Classes to store data for an experiment, but does not include a Data Class definition for another object, such as *Image*, that is also critical for the experiment. New Data Classes must be created from scratch; with details such as the name, description, and attribute properties explicitly defined using the web interface (Figure 16). Although the researchers themselves understand their data the best, this task can still be challenging without any data modeling background. A particularly tricky aspect of creating a Data Class from scratch is selecting the data types for each attribute. An understanding of the basic database types and of relational links between Data Classes may require a substantial learning curve.



The Eye Lab

Cataract ID Study

Create New Data Class

Data Class Name:	Condition
Description:	Condition of an animal subject
Show on Data Store Home Page:	☑

Define Class Attributes

Enter up to 15 Attributes, then press Submit

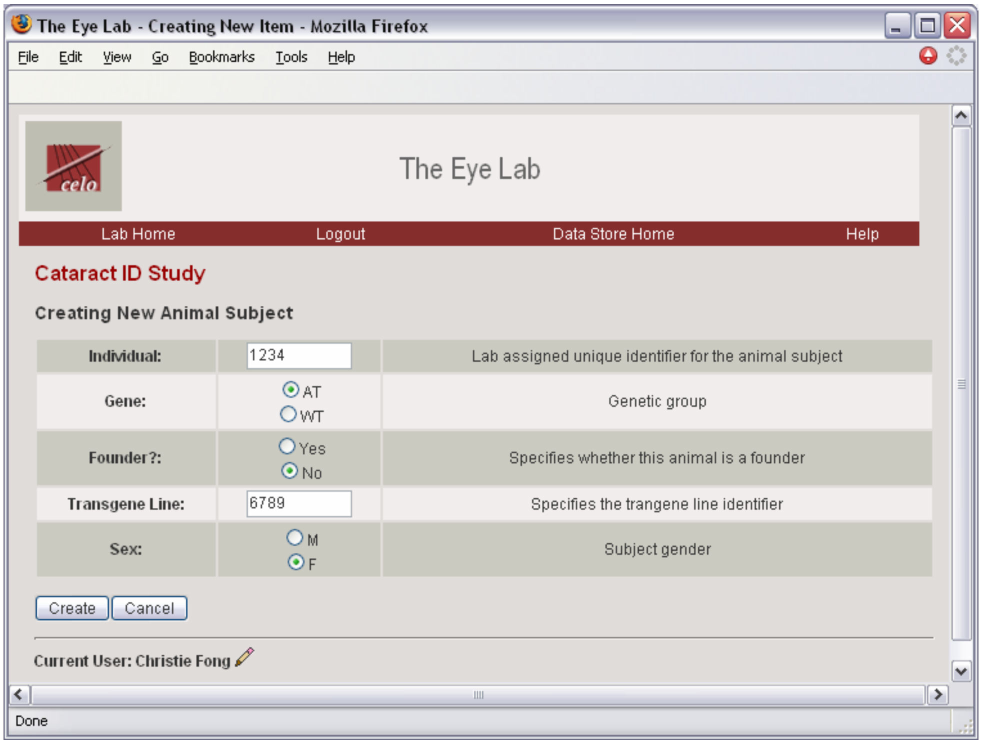
Attribute Num	Name	Required	Unique	Label	Type	Max Length	Option List
1	Drug	☑	□	□	Limited length string		
2		□	□	□			
3		□	□	□			

Figure 16. Create New Data Class

4.8 Enter Items

After a laboratory is satisfied with the Data Classes that represent a set of research data, items can be entered into the database. CELO provides two different interfaces for entering items: an interface for entering a single item and an interface for entering multiple items. To enter multiple items at once, the user specifies the number of items to add and a web form similar to a spreadsheet format is automatically generated. Entering multiple items at once can help improve the efficiency of data entry.

The web forms generated for entering data are composed using the properties of each attribute for the specified Data Class. A combination of the data type and the specified *edit widget* for each attribute determine how the input fields are displayed to the user. For example, if the data type is `enum('M','F')` and the widget specified is `radio_group`, then a group of two radio buttons with values *M* and *F* is displayed (Figure 17).



Cataract ID Study

Creating New Animal Subject

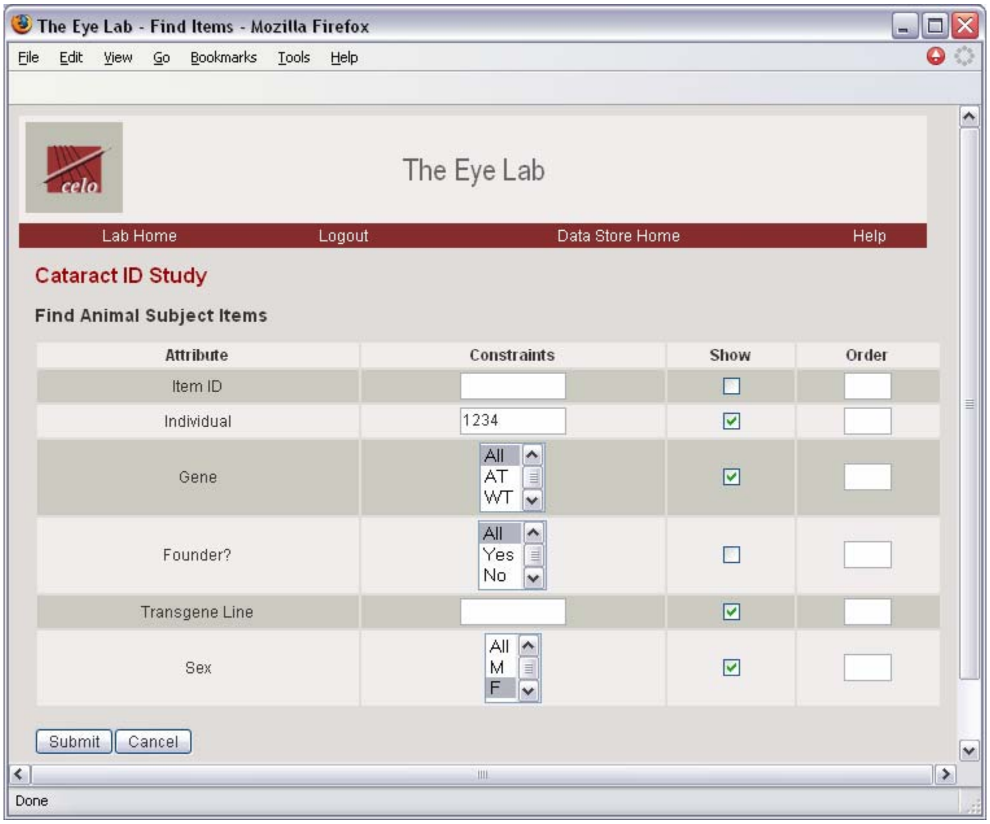
Individual:	1234	Lab assigned unique identifier for the animal subject
Gene:	● AT ○ WT	Genetic group
Founder?:	○ Yes ● No	Specifies whether this animal is a founder
Transgene Line:	6789	Specifies the trangene line identifier
Sex:	○ M ● F	Subject gender

Current User: Christie Fong

Figure 17. Create New Item

4.9 Query for Items

It can be a challenge, especially with large data sets, for researchers to easily and efficiently find particular research data items. A critical feature of laboratory informatics solutions is therefore the ability to query for data. CELO utilizes relational database technology which benefits from powerful SQL queries. Two interfaces are provided to users for generating queries into the database. The simple query interface is a simple web form that allows users to specify attribute constraints for finding items belonging to a single Data Class (Figure 18). Similar to the create item web forms, this interface is constructed using the attribute properties of the Data Class. The interface hides the complexities of the SQL query statements which can be time consuming to construct and difficult for novice database users to understand.


The Eye Lab - Find Items - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
celo
Lab Home
Logout
Data Store Home
Help
Submit
Cancel
Done

The Eye Lab

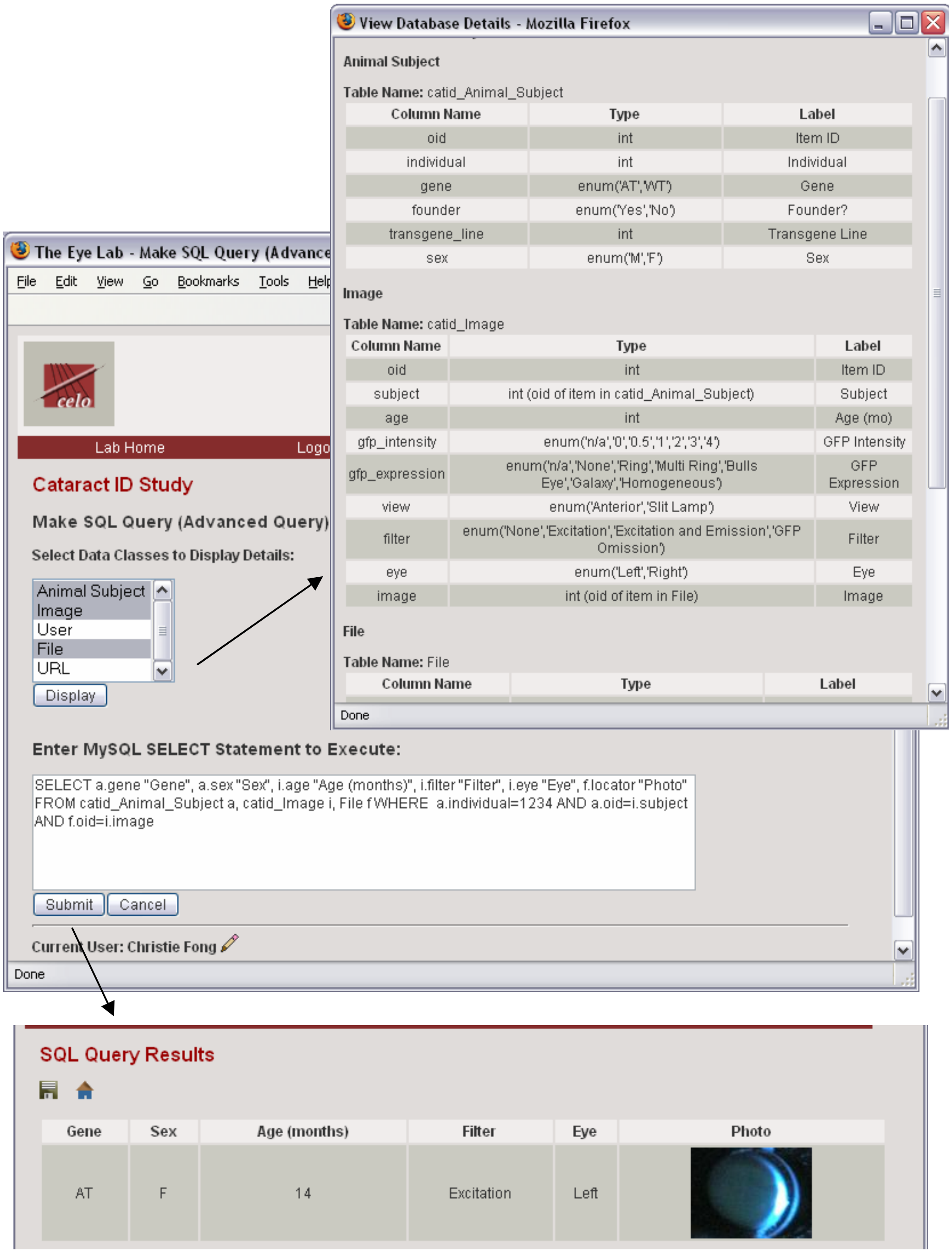
Cataract ID Study

Find Animal Subject Items

Attribute	Constraints	Show	Order
Item ID		□	
Individual	1234	☑	
Gene	All AT WT	☑	
Founder?	All Yes No	□	
Transgene Line		☑	
Sex	All M F	☑	

Figure 18. Simple Query Interface

The simple query interface may be an efficient and user friendly method for finding research data, but the types of queries it can construct is very limited. In order allow researchers to perform more flexible queries, an advanced query interface is also available. This interface provides tools for viewing the underlying database schema that is associated with Data Classes such that researchers familiar with database programming can directly compose SQL SELECT statements (Figure 19). This querying method can generate more complex data views that can, for example, combine data from multiple Data Classes or return the results of various mathematical operations on certain data values.


View Database Details - Mozilla Firefox
The Eye Lab - Make SQL Query (Advance
File Edit View Go Bookmarks Tools Help
celo
Lab Home
Logo
Display
Submit
Cancel
Done

Animal Subject

Table Name: catid_Animal_Subject

Column Name	Type	Label
oid	int	Item ID
individual	int	Individual
gene	enum('AT','WT')	Gene
founder	enum('Yes','No')	Founder?
transgene_line	int	Transgene Line
sex	enum('M','F')	Sex

Image

Table Name: catid_Image

Column Name	Type	Label
oid	int	Item ID
subject	int (oid of item in catid_Animal_Subject)	Subject
age	int	Age (mo)
gfp_intensity	enum('n/a','0','0.5','1','2','3','4')	GFP Intensity
gfp_expression	enum('n/a','None','Ring','Multi Ring','Bulls Eye','Galaxy','Homogeneous')	GFP Expression
view	enum('Anterior','Slit Lamp')	View
filter	enum('None','Excitation','Excitation and Emission','GFP Omission')	Filter
eye	enum('Left','Right')	Eye
image	int (oid of item in File)	Image

File

Table Name: File

Column Name	Type	Label

Cataract ID Study

Make SQL Query (Advanced Query)

Select Data Classes to Display Details:

- Animal Subject
- Image
- User
- File
- URL

Enter MySQL SELECT Statement to Execute:

```
SELECT a.gene "Gene", a.sex "Sex", i.age "Age (months)", i.filter "Filter", i.eye "Eye", f.locator "Photo"
FROM catid_Animal_Subject a, catid_Image i, File f WHERE  a.individual=1234 AND a.oid=i.subject
AND f.oid=i.image
```

Current User: Christie Fong

SQL Query Results

Gene	Sex	Age (months)	Filter	Eye	Photo
AT	F	14	Excitation	Left	

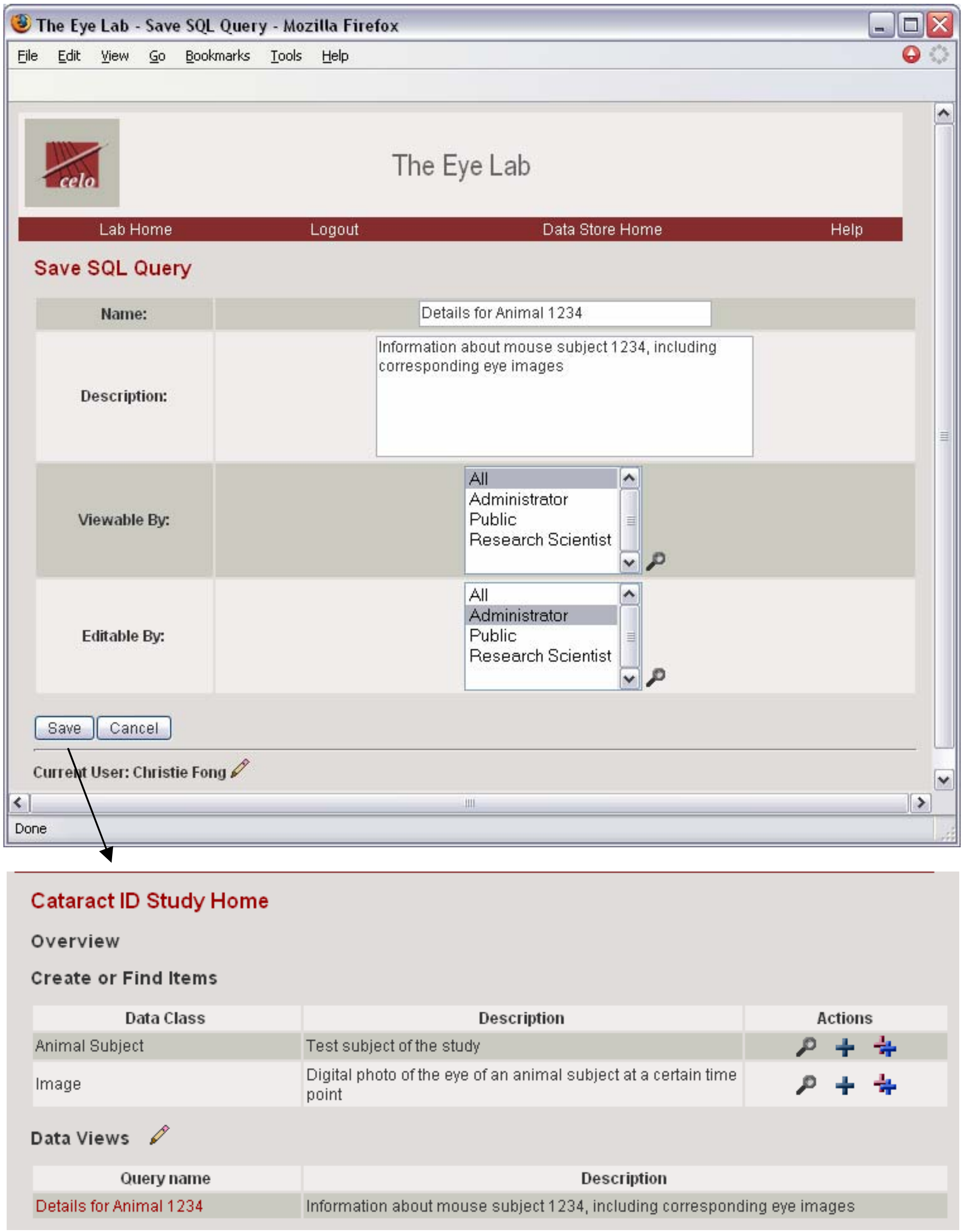
Figure 19. Advanced Query Interface

When a SQL statement is submitted, the system displays the results in a formatted web table. Special processing is used for displaying values for columns with certain reserved names. A column name of `oid`, for example, will not display the actual oid value, but instead a link to the item with the corresponding oid value.

The two different querying methods, simple and advanced, were developed in an attempt to provide the end users with options of varying balances of the trade-offs between usability and flexibility.

4.10 Save a Query

Researchers may find that certain queries need to be performed frequently. These queries can be saved for future access such that they do not need to be recreated each time they need to be run. Users can save a query from the query results page by simply selecting the “floppy disc” save icon that should be familiar to most computer users. The user will be prompted to specify a query name, description, and User Group permissions for viewing and editing the query. Saved queries are displayed on the Data Store’s Home Page as a link that can be easily selected to execute the query (Figure 20).


The Eye Lab - Save SQL Query - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
celo
The Eye Lab
Lab Home
Logout
Data Store Home
Help
Save
Cancel
Done

Save SQL Query

Name:	Details for Animal 1234
Description:	Information about mouse subject 1234, including corresponding eye images
Viewable By:	All (selected) Administrator Public Research Scientist
Editable By:	All Administrator (selected) Public Research Scientist

Current User: Christie Fong

Cataract ID Study Home

Overview

Create or Find Items

Data Class	Description	Actions
Animal Subject	Test subject of the study	
Image	Digital photo of the eye of an animal subject at a certain time point	

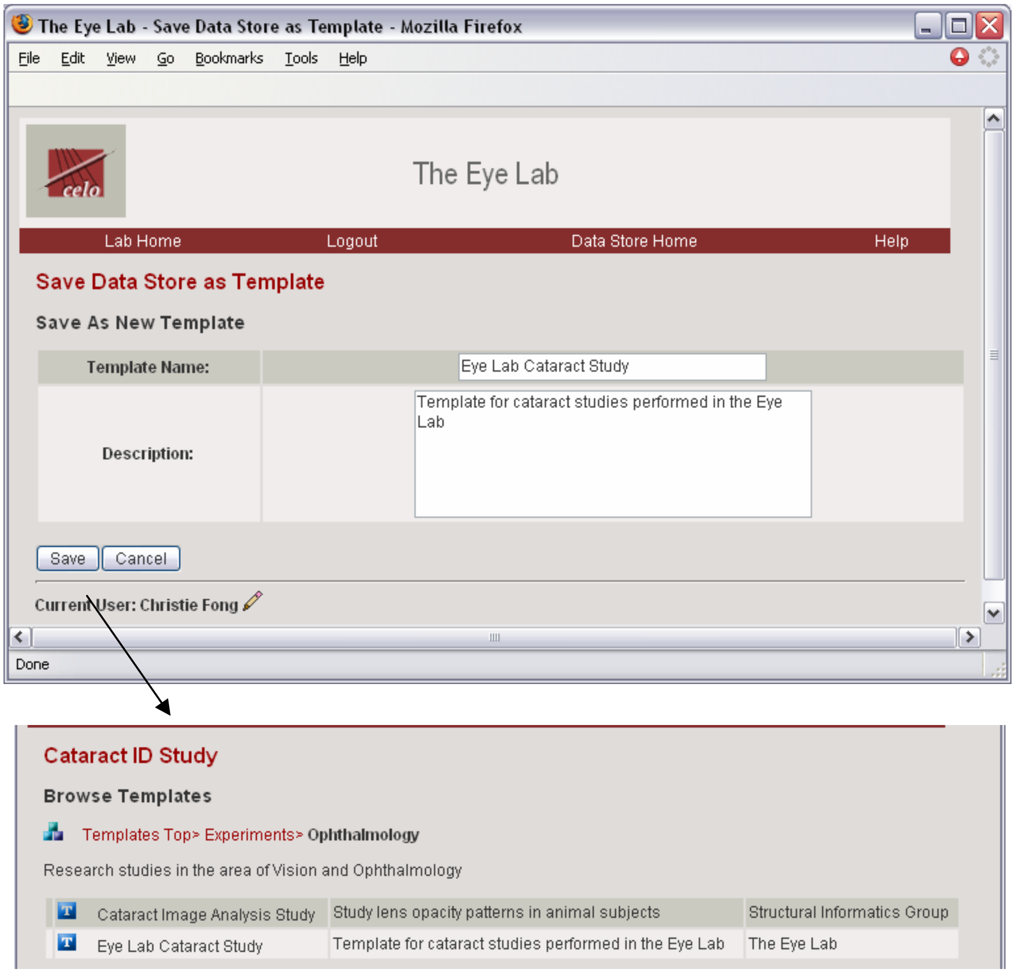
Data Views

Query name	Description
Details for Animal 1234	Information about mouse subject 1234, including corresponding eye images

Figure 20. Save a Query

4.11 Save a Template

Laboratories may want to reuse the structure of a Data Store that captures the data of a research area effectively. A lab might, for example, perform multiple different experiments which generate very similar types of data sets. In this scenario, researchers can save the data representations they have defined for one experiment and then efficiently regenerate the same representation for a similar experiment. Modifications to the template representation can be easily made as necessary. A laboratory might also want to share a Data Store definition with other laboratories performing similar research in order to encourage consistency or simply to help fellow researchers. Laboratories can accomplish these tasks by saving the Data Class and Query definitions of a Data Store as an XML template. Logged in laboratory Administrators can save a Data Store as a template through a link provided in the tools section of the Data Store Home Page. The Administrator will be prompted for a template name and description. Other template details, such as the date created, author, and authoring organization will be automatically generated. Once the template has been created, the user can then select the appropriate template collection into which to place the new template, with an option to create a new collection. Any other laboratory with access to the CELO server can then view the new template details and use the template to assist in the definition of Data Stores (Figure 21).


The Eye Lab - Save Data Store as Template - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
celo
The Eye Lab
Lab Home
Logout
Data Store Home
Help
Save
Cancel
Done

Save Data Store as Template

Save As New Template

Template Name:	Eye Lab Cataract Study
Description:	Template for cataract studies performed in the Eye Lab

Current User: Christie Fong

Cataract ID Study

Browse Templates

Templates Top> Experiments> Ophthalmology

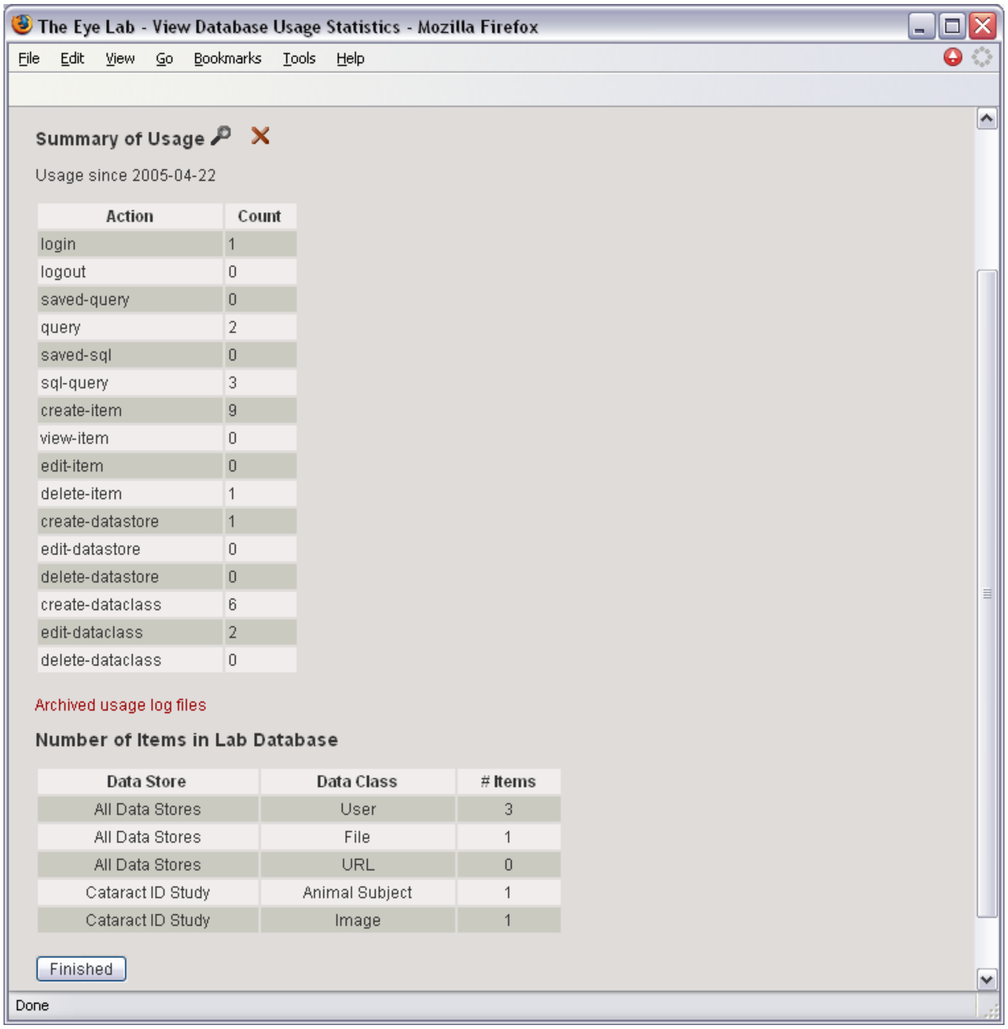
Research studies in the area of Vision and Ophthalmology

Cataract Image Analysis Study	Study lens opacity patterns in animal subjects	Structural Informatics Group
Eye Lab Cataract Study	Template for cataract studies performed in the Eye Lab	The Eye Lab

Figure 21. Save a Template

4.12 View the Usage Log

Another useful feature of the CELO system is the ability to monitor activity within the laboratory system. This tool helps labs track which system features are being used, how frequently they are being used, and which users are using them. Messages are logged to a usage log each time a particular action is performed. Users with permissions to view the usage log can access the log using a link provided in the tools section of the Laboratory Home Page. A *summary of usage* table lists the number of times each of the monitored actions has been performed and gives an idea of how often the various features of the system are being used. The number of items currently entered in the database by Data Class is also provided and helps researchers determine the size of their database.


The Eye Lab - View Database Usage Statistics - Mozilla Firefox
File
Edit
View
Go
Bookmarks
Tools
Help
Finished
Done

Summary of Usage

Usage since 2005-04-22

Action	Count
login	1
logout	0
saved-query	0
query	2
saved-sql	0
sql-query	3
create-item	9
view-item	0
edit-item	0
delete-item	1
create-datastore	1
edit-datastore	0
delete-datastore	0
create-dataclass	6
edit-dataclass	2
delete-dataclass	0

Archived usage log files

Number of Items in Lab Database

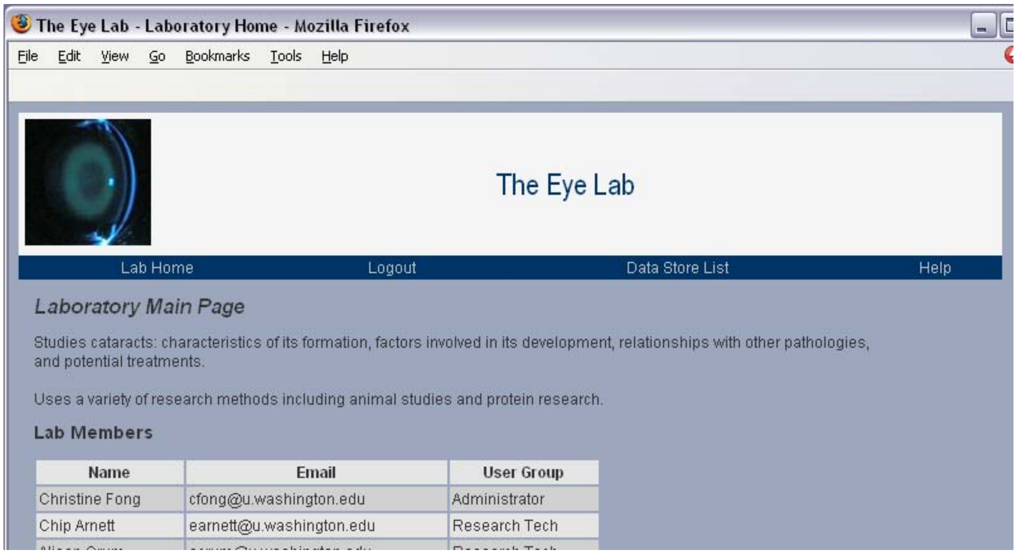
Data Store	Data Class	# Items
All Data Stores	User	3
All Data Stores	File	1
All Data Stores	URL	0
Cataract ID Study	Animal Subject	1
Cataract ID Study	Image	1

Figure 22. View Usage Log

Advanced features of the usage log include a method for querying for log entries. The user can specify, for example, a date range, a User ID, or a subset of actions to find usage messages for. The query results can be saved as an HTML formatted Usage Report that can be accessed through a Usage Archives page for future reference. Another useful feature is the ability to clear the usage log. When clearing the usage log, the user is given the option to save the log to a comma separated value file before clearing it. These logs saved to file can be viewed or downloaded through the Usage Archives page for future reference. The new log start date is displayed with the usage summary statistics such that users can more accurately estimate system activity using the message counts.

4.13 Customizing Web Display Settings

The web page for every laboratory will, by default, use the same coloring scheme and font styles. Each laboratory, however, might want to customize these display settings in order to more easily distinguish its web page from other labs. In order to do this, a user with permissions to edit the display settings must simply use the configuration tools to specify new color codes and font names and sizes. Laboratories can also choose to replace the default CELO logo displayed in the banner at the top of each web page with a unique lab logo (Figure 23).


The Eye Lab - Laboratory Home - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
The Eye Lab
Lab Home
Logout
Data Store List
Help

Laboratory Main Page

Studies cataracts: characteristics of its formation, factors involved in its development, relationships with other pathologies, and potential treatments.

Uses a variety of research methods including animal studies and protein research.

Lab Members

Name	Email	User Group
Christine Fong	cfong@u.washington.edu	Administrator
Chip Arnett	earnett@u.washington.edu	Research Tech

Figure 23. Customized Web Page Display Settings

4.14 Log Out

To complete a session using the system, a user logs out using the *Log Out* link on the web page navigation bar. The user will no longer have access to the features of the system defined by the user's User Group permissions. Any further activity within the system will be considered to be performed by a generic Public user. Features that are available to this user are determined by the permissions assigned to the Public User Group which, by default, only include read access features. Web links to features for adding new items into the database, for example, will be grayed out. This functionality gives labs some control over what aspects of their database are available to the general public.

Chapter 5: Evaluation

CELO was designed with consideration of the nine system requirements that we identified based on our review of existing informatics solutions and our experiences building laboratory data management systems as discussed in Chapter 2. Our hypothesis is that CELO is able to efficiently build data management systems that meet the specific needs of biomedical laboratories. In order to test this hypothesis, we have used CELO to recreate existing systems that we have already built for particular research labs. The advantage of this evaluation approach is that we have a solid understanding of the needs of these research labs, we can easily identify system features that are most valuable to the labs, and we can compare the development times of these features between the original and recreated systems. Furthermore, we can perform the evaluation without adversely impacting the labs by needing to get extensive input from them. For each of the three laboratories we have previously worked with that we used in our evaluation, we provide a brief description of the lab's research efforts and its specific informatics needs. For each of the major features of the original systems, we discuss how well we were able to recreate the feature using CELO and discuss any notable differences in development time. In our evaluation, we also assess how important each of the nine system requirements is for the systems being recreated. This assessment both helps to validate our selection of core requirements and also helps indicate whether CELO is an appropriate informatics solution for each lab.

We recognize that the design decisions we made when developing CELO may have been biased due to our experiences working on informatics solutions for particular labs. It is therefore important to evaluate the generalizability of the system by creating data management systems for laboratories we have not yet worked with. We describe the research focus and data management needs for one such lab, and describe how CELO was used to generate an informatics solution. Our evaluation includes a discussion of the needs that were met, the needs that were unable to be met, and the efficiency with which the features were implemented. Although an evaluation of one additional laboratory provides some insight on the generalizability of the system, we recognize that a much larger scale evaluation needs to be performed. We therefore also describe our future evaluation plans and how these plans will help drive improvements for future versions of the system.

5.1 The Eye Lab Image Repository

The Eye Lab is a lab in the University of Washington's Department of Biological Structure led by Dr. John Clark. The lab performs research on factors affecting the development of cataracts, one of the leading causes of blindness in the world. Some experimental studies performed by the lab involve examining the progressive changes in lens opacification in transgenic mice[48, 49]. In order to analyze the spatial and temporal variations in lens phenotype, enormous sets of digital images and related data are collected. Organizing and analyzing such large numbers of mouse eye images becomes a challenging and time consuming task[22].

In order to address the data management needs of the lab, we developed a web-based image repository using the WIRM toolkit described earlier. We worked closely with the lab in order to design the database schema for representing experiment data. Multiple custom Perl scripts were developed on top of the existing WIRM framework to implement features to help the Eye Lab researchers organize and analyze their image data. The Eye Lab has used the system to store data for three experiments, including over a thousand images and profiles for over a hundred mouse subjects. We were able to use CELO to recreate several of the features of this original image repository. One of the original system's more unique features developed by request of the lab members could not be recreated, demonstrating the importance of the plugins for customization system requirement that CELO does not currently satisfy.

Controlling User Access

The Eye Lab members wanted the data in the lab database to be public such that collaborators and others interested in the lab's research could easily view research data without having to log in to the system. They also, however, wanted to ensure that only researchers of the Eye Lab would be able to add or edit items in the database. In order to implement this functionality, the original Eye Lab image repository utilized the WIRM framework for controlling user access. This framework used a pre-defined set of user groups that users were to be assigned to. User group access to system functions was then hard-coded into each custom script. CELO, on the other hand, uses a configurable permissions system that allow users to define their own user groups and assign a set of accessible system features to each user group. CELO is therefore able to control user access like the original system while providing users with more flexibility to modify the permissions.

Organizing Data from Separate Experiments

The Eye Lab must manage data from multiple experiments. In order to help the researchers organize the data from these separate experiments, we designed an *Experiment* object for the original Eye Lab Image Repository. Separate sets of data can be created for each *Experiment,*and researchers can access the data entered for each experiment through different web pages. For example, users can access data for the “Huntington Mice Study” and the “Cataract ID” study through separate dedicated web pages.

The *Experiment* object defined for the original Eye Lab system is equivalent to the *Data Store* object of CELO. Both are used to organize related data into separate groups, such that adding, viewing, and finding data is not complicated by the clutter of non-related data. The Experiment object was defined by an experiment name, hypothesis, researcher list, and date range. Data Stores, on the other hand, are more generic and are only defined by a name and description. While less descriptive, Data Stores allow laboratories to create groups of data to define experiments as well as for other purposes. For example, a Data Store can also represent an electronic library for laboratory publications. An advantage of the more detailed Experiment representation is it would make features involving retrieval or organization by experiment properties such as hypothesis or date easier to implement. Features such as this, however, are not currently in our plans for future work.

Defining Representations of Data

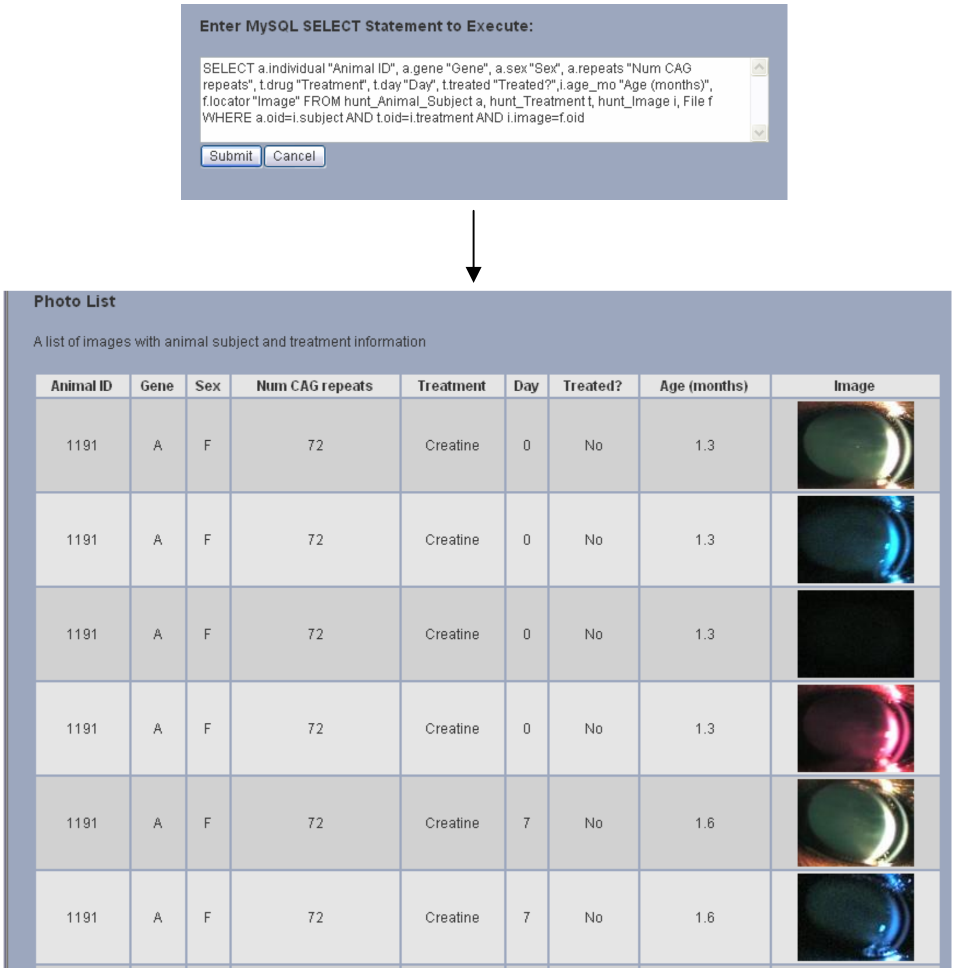
The original Eye Lab Image Repository provided the researchers with interfaces for defining custom data types. The data types that the Eye Lab members created for representing three experiments were very similar to each other, with similar definitions for objects like Treatment, Animal Subject, and Image. We used these data types to define one of the default XML templates for the CELO system. Recreating the representations for the three experiments, therefore, simply required selecting this template and making minor modifications to the generated Data Classes. Although CELO includes a web interface similar to the original Eye Lab system for defining data types, using the template was much easier and more efficient for this task.

Creating and Viewing Items

A critical feature of the original Eye Lab Image Repository is its support for adding image files into the database. The system allows researchers to use their web browsers to upload mouse eye image files that they need to analyze along with properties related to each image, such as the age of the mouse in the photo and the type of camera filter used for the photo. The system stores and organizes the files on the server and allows users to retrieve images based on image properties. When the user views an image item, the system processes the database entries to display the image thumbnail along with image property details. Because the CELO system also uses the WIRM code base, it can utilize the WIRM features for handling image files and therefore is able to provide similar support for creating and viewing the Eye Lab images.

Finding Images

Being able to quickly find particular mouse eye images is a valuable feature of the original Eye Lab image repository. The system provides a web interface in which users can simply specify certain image property constraints; for example, images for mice at 2 months of age. The system will return a table listing matching results, displaying image thumbnails along with associated image properties. CELO provides a very similar querying interface to the original system. It also, however, provides a more powerful querying feature that allows users to directly construct SQL queries. The feature can generate views of data that join information from multiple tables, such as a list of images along with associated animal subject and treatment data. (Figure 24).



Enter MySQL SELECT Statement to Execute:

```
SELECT a.individual "Animal ID", a.gene "Gene", a.sex "Sex", a.repeats "Num CAG
repeats", t.drug "Treatment", t.day "Day", t.treated "Treated?",i.age_mo "Age (months)",
f.locator "Image" FROM hunt_Animal_Subject a, hunt_Treatment t, hunt_Image i, File f
WHERE a.oid=i.subject AND t.oid=i.treatment AND i.image=f.oid
```

Photo List

A list of images with animal subject and treatment information

Animal ID	Gene	Sex	Num CAG repeats	Treatment	Day	Treated?	Age (months)	Image
1191	A	F	72	Creatine	0	No	1.3	
1191	A	F	72	Creatine	0	No	1.3	
1191	A	F	72	Creatine	0	No	1.3	
1191	A	F	72	Creatine	0	No	1.3	
1191	A	F	72	Creatine	7	No	1.6	
1191	A	F	72	Creatine	7	No	1.6	

Figure 24. Eye Lab Image Repository Advanced Query

Creating Image Matrices for Image Comparison and Analysis

The Eye Lab researchers probably consider the most valuable feature of the original image repository to be its ability to automatically generate the image matrices for comparing images. A simple user interface allows users to specify image constraints and to select which image fields to use as the columns and rows of the image matrix. We were not able to recreate this feature using CELO. The inability of the CELO-generated system to provide this valuable lab-specific feature demonstrates the importance of the plugins for customization requirement. The ability to integrate a custom-built feature into the base CELO system would greatly improve the utility of the system.

5.2 The Cortical Stimulation Mapping Database

Dr. George Ojemann of the University of Washington's Department of Neurological Surgery leads several studies for mapping language related areas of the brain using direct cortical stimulation in patients undergoing awake brain surgery. These cortical stimulation mapping (CSM) studies are performed on patients during presurgical treatments for intractable epilepsy. Electrical current is applied directly to areas of the cortex, and patients are observed while performing various language related tasks in order to determine language related areas[63-65].

We developed the CSM Database to help the researchers manage data about patients, surgeries, study trials, and brain images. The interface of the system was also designed to facilitate experiment workflow management, directing users through a series of interfaces for viewing or entering data. The system navigates the user through viewing or creating a patient, to the patient's associated surgeries of imaging studies, to various data associated with these objects, to finally generate enough information to be used as input to a custom brain visualization application. The original CSM Database was developed over the course of seven years using the WIRM toolkit as a base. Using CELO, we were able to partially recreate many of the major features of the original system very efficiently. We were unfortunately only able to evaluate the viewing and retrieval features of the original system because we were unable to obtain system editing privileges due to the sensitive nature of the data being stored in the database. Our comparison of the original and recreated systems based only on the viewing and retrieval features still illustrate many of the strengths and weaknesses of the CELO system.

Controlling User Access

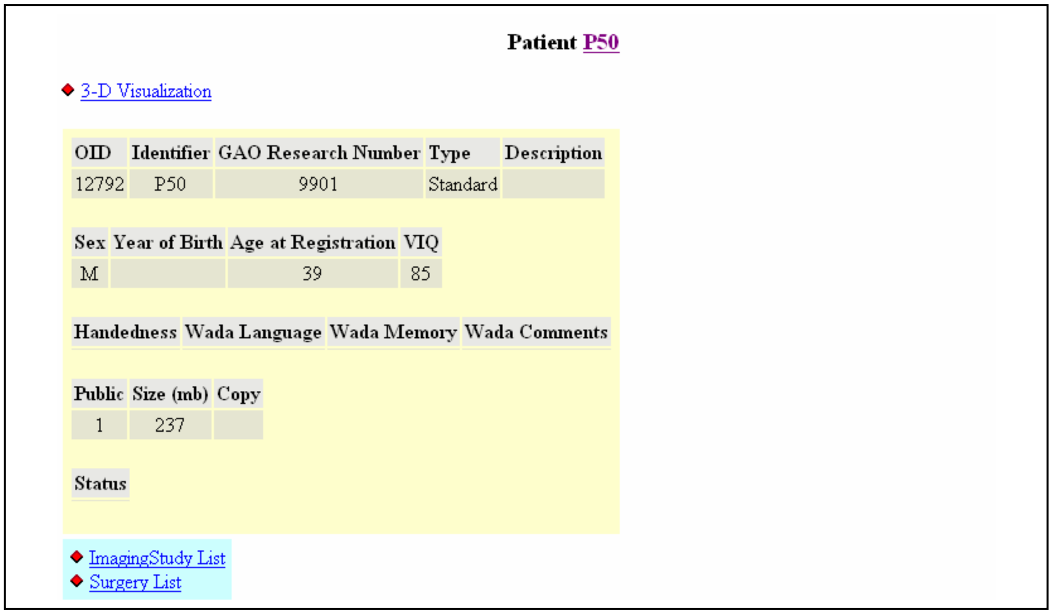
The CSM study has unique security needs due to the sensitive information stored in the database about human patients. Although identifying data such as name and date of birth were intentionally not included in the database, the researchers wanted to ensure confidentiality by allowing only some system users to have access to database tables with sensitive information. The original CSM database, like the Eye Lab image repository, implements user access control to particular features by hard coding User Group checks in the appropriate places of every CGI script. CELO, on the other hand, allows researchers to configure permissions using a web tool offering a pre-defined set of access options, such as viewing or editing items. The drawback of this customization feature is that the control of access permissions is not as flexible. CELO, therefore, was unable to recreate the unique user access control of the original CSM database. Because the original system hard-coded permissions into its scripts, there was more flexibility in the types of features to control access to.

Defining Representations of Data

Research data are represented using about 30 interrelated tables in the original CSM database, representing diverse object types from patients to cortical stimulation sites. We created one of the default CELO templates based on this existing database schema. Defining representations of data for the recreated CSM database system was therefore as simple as selecting the appropriate template. All the tables and necessary relationships were automatically generated, and the web interfaces for creating, editing and querying items were immediately ready for use. Use of the template feature to define research data representations was effective, simple, and efficient.

Creating and Viewing Items

Many custom user interfaces were developed for the original CSM database. For example, a web interface for viewing details about a patient displayed only a subset of patient properties and grouped together other properties into distinct rows. This formatting was specifically designed such that researchers could easily scan the page to comprehend the patient data. CELO was able to capture some aspects of the interface customizations, such as displaying user friendly labels for attributes corresponding to database table columns. The interface configuration options, however, were unable to recreate some of the valuable characteristics of the original interfaces. The generic two column table for displaying attribute values in the CELO-generated system, for example, is more difficult to quickly scan than the original interface (Figure 25).



Patient P50

- 3-D Visualization

OID	Identifier	GAO Research Number	Type	Description
12792	P50	9901	Standard	

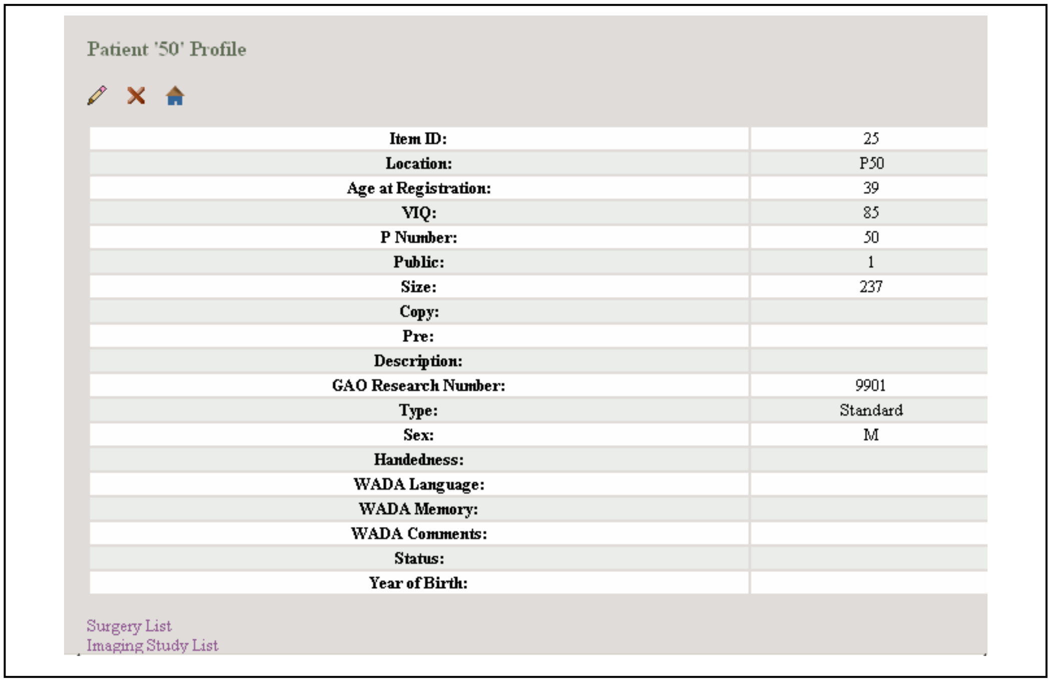
Sex	Year of Birth	Age at Registration	VIQ
M		39	85

Handedness	Wada Language	Wada Memory	Wada Comments

Public	Size (mb)	Copy
1	237	

Status

- ImagingStudy List
- Surgery List



Patient '50' Profile

Item ID:	25
Location:	P50
Age at Registration:	39
VIQ:	85
P Number:	50
Public:	1
Size:	237
Copy:	
Pre:	
Description:	
GAO Research Number:	9901
Type:	Standard
Sex:	M
Handedness:	
WADA Language:	
WADA Memory:	
WADA Comments:	
Status:	
Year of Birth:	

Surgery List
Imaging Study List

Figure 25. CSM Database Comparison for Viewing Patient Item

Our evaluation has also identified several other interfaces which CELO could not recreate. One example is the list of stimulation sites that is displayed for a given surgery. The original CSM database provides details for each surgical stimulation site such as the site label, associated grid stimulation sites, and whether the site has any corresponding trials. Retrieving details such as these for the stimulation site list requires a set of custom queries. The equivalent CELO stimulation site list only displays the site label and cannot be configured to show additional information (Figure 26).

Access to Data

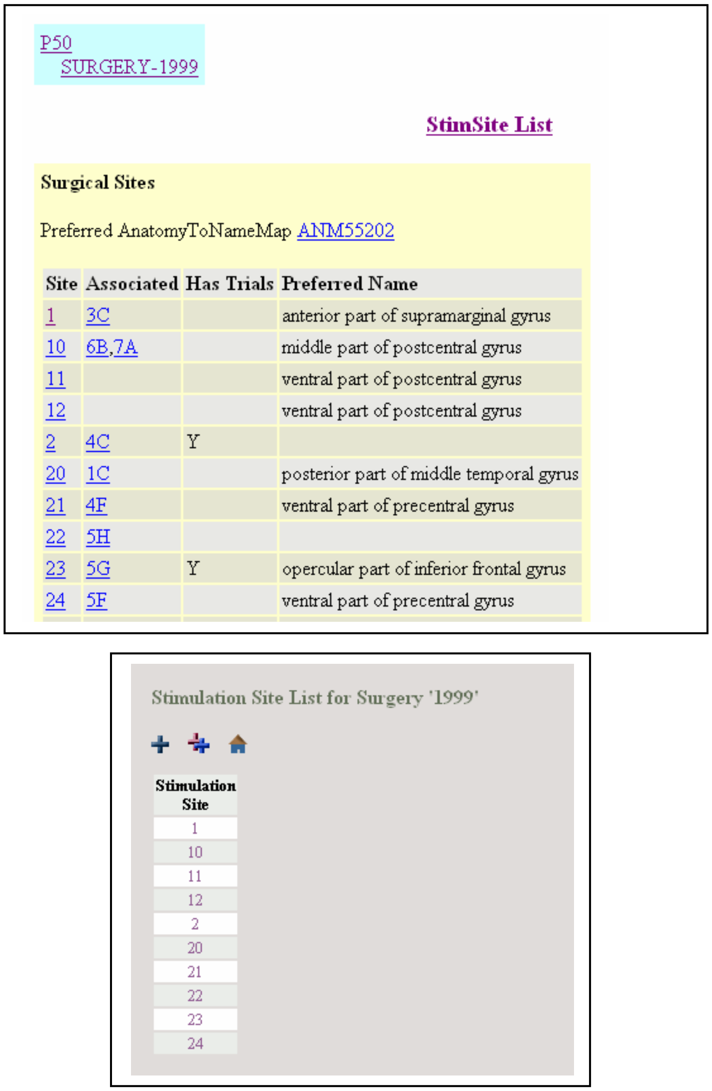
The main menu of the original CSM database contains links to webpages in which users can browse a list of study patients. The *Patient List* page displays patient information such as the patient ID, type, research number, exam number, age, sex, and verbal IQ (VIQ). One complexity of the Patient List is that users not logged in to the system (Public users) can only view a subset of the patients in the database. An equivalent to the Patient List can be easily created in the CELO-generated system using the advanced query feature. The feature allows users to construct a SQL statement for retrieving patient data joined with data from related tables such as Exam. The SQL query can be saved and made viewable only to particular user groups. A similar SQL query that returns only public patients (`public` attribute set to 1) can also be created and saved for only Public users to view. The Patient List feature of the CSM database can therefore be effectively recreated using the CELO web tools.

A second link is also available from the CSM database main menu for viewing a more extensive set of data related to each patient. The *Patient Status* page provides the same details as the Patient List along with additional information such as the number of scenes, study trials, and photos associated with each patient. This detailed list of information must be constructed using the results from multiple related queries. CELO only supports displaying results returned from a single SQL query and therefore cannot recreate the Patient Status page.

System Navigation

Navigation through the original CSM database begins with the list of patients which links to items associated with a particular patient, such as surgeries and imaging studies, continues with data associated with these objects, and so on and so forth. This drilldown design for viewing items in the database allows users to navigate through the system in a logical manner. A similar drilldown navigation scheme was constructed in CELO by setting options such that data access must begin with a Patient item, with other items accessible only through links generated through Data Class relationships.

Although CELO can recreate the CSM database functionality for navigating down the object hierarchy, it does not support the same functionality for navigating back up the hierarchy. The original system, for example, provides links for users to easily navigate from a stimulation site list back up to either the patient or surgery item. The CELO interface, on the other hand, does not provide links to return to the patient or surgery, nor does it even indicate the patient from which the stimulation site originated. This makes navigation through the system more difficult and confusing for users (Figure 26).



P50
SURGERY-1999

StimSite List

Surgical Sites

Preferred AnatomyToNameMap ANM55202

Site	Associated	Has Trials	Preferred Name
1	3C		anterior part of supramarginal gyrus
10	6B,7A		middle part of postcentral gyrus
11			ventral part of postcentral gyrus
12			ventral part of postcentral gyrus
2	4C	Y	
20	1C		posterior part of middle temporal gyrus
21	4F		ventral part of precentral gyrus
22	5H		
23	5G	Y	opercular part of inferior frontal gyrus
24	5F		ventral part of precentral gyrus

Stimulation Site List for Surgery '1999'

Stimulation Site
1
10
11
12
2
20
21
22
23
24

Figure 26. CSM Database Comparison for Stimulation Site List

Visual Brain Mapper Application

In addition to all the specialized navigation features and interfaces, the CSM database contains several other features unique to the CSM studies. One of these features is the integration of the Visual Brain Mapper (VBM) application[19]. This application requires an input file that is generated from a very specific set of data with certain value restrictions that have been created for a patient. The VBM uses the input file to generate a set of brain scenes and maps for the patient. The CSM database coordinates the various steps required for successful execution of the VBM. Such complex and domain specific functionality can only be accomplished through the creation of custom scripts. Support for the VBM is not generalizable and would therefore not be an appropriate feature for a system like CELO. The value of being able to integrate custom scripts for supporting the VBM in the CSM database, however, demonstrates how critical the system requirement for plugins can be.

5.3 Single Unit Recording Database

Dr. George Ojemann also works closely with Dr. David Corina of the University of Washington's Department of Neurological Surgery to perform Single Unit Recording (SUR) studies. Similar to the CSM studies, data are collected during awake neurosurgery in patients undergoing surgical treatment for epilepsy. In contrast to the CSM studies, the SUR studies use microelectrodes to measure extracellular recordings of single neurons in the brain. The electrical activity of different neurons is monitored as the patients are given a series of tasks. Such studies attempt to identify neurons that participate in various cognitive tasks. Explicit memory, for example, is studied by presenting the patients with a number of trials testing recent memory of auditory words, nameable object pictures, or text words[63, 64, 66-68].

A current biomedical informatics graduate student, Hao Li, has developed a prototype system to manage SUR study data including details about patients, surgeries, protocols, trials, neurons, and electrodes. Various data files, such as timestamp files and neuron firing files are also managed and associated with related data. The prototype SUR Experiment Management System (EMS) was created using a system under development that automatically transforms an ontological representation of research data into a web-based relational database system[69]. The prototype SUR Experiment Management System was designed in close collaboration with the researchers performing the SUR studies. Using CELO, we were able to recreate the majority of the features of the prototype system with some minor user interface differences.

Defining Representations of Data

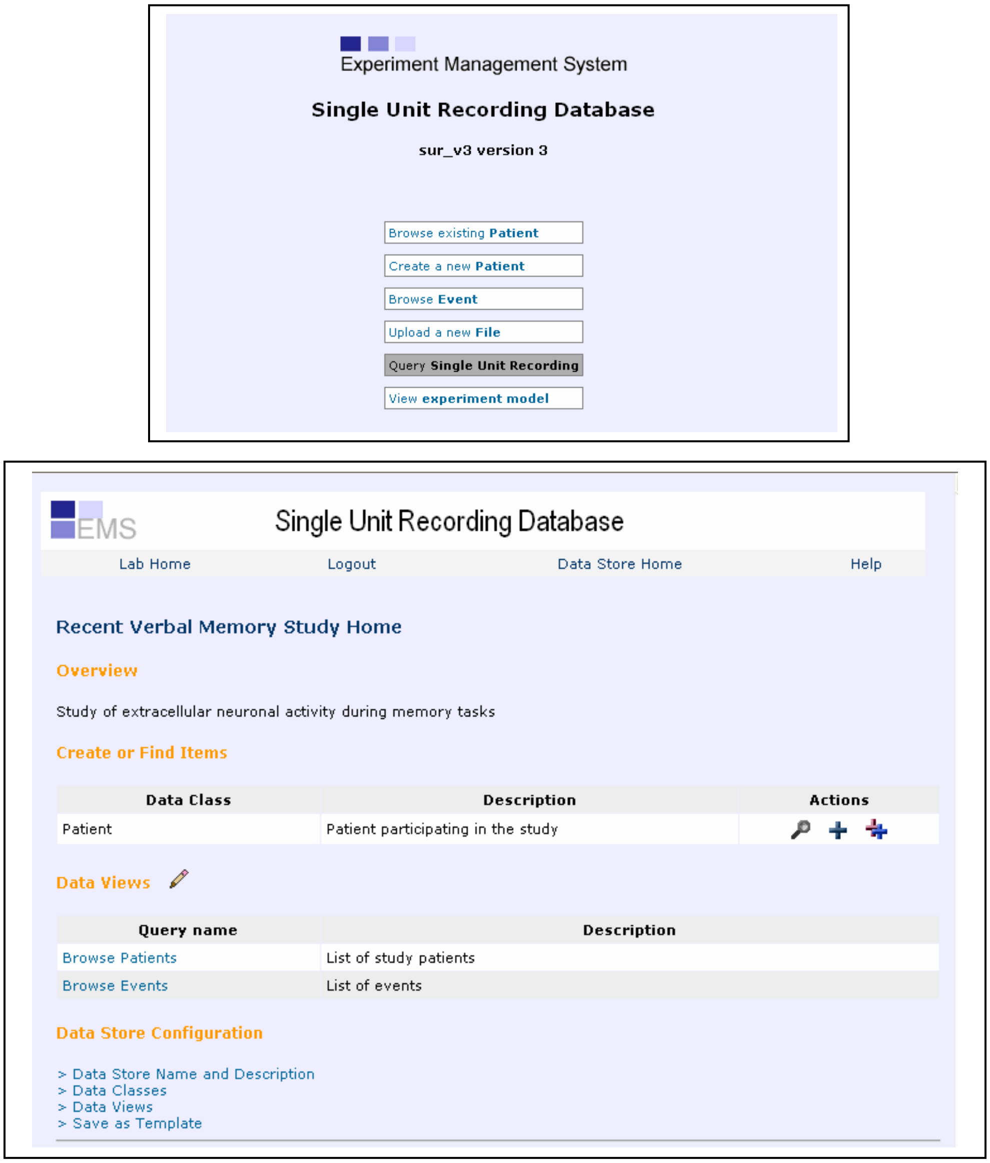
The original SUR database schema was developed with a thorough understanding of the research by working directly with the end users, the researchers themselves. As with the Eye Lab image repository and the CSM database, we used the existing schema to create a default template for single cell recording studies. Defining database representations for the SUR data in our recreated system, therefore, simply required selecting this template to automatically generate the appropriate tables and relationships. This demonstrates again how the CELO template system can provide an easy and efficient method for defining data representations.

Access to Data

Access to data in the original prototype SUR Experiment Management System begins at the main menu. The main menu contains links to browse existing patients, create a new patient, browse existing events, and to view the experiment model. The functions of each of these links were easily recreated using the configuration options of the CELO system. For example, a link to browse all of the existing patients was created by constructing a simple query for Patient items, specifying no query constraints. Although the features of the original SUR system's main menu can be easily recreated using CELO, a minor drawback of the recreated version is that the data access options are not explicitly listed in a menu format and therefore may be more confusing to users (Figure 27).

Creating and Viewing Items

As noted earlier, the features for browsing the list of Patients and Events differs slightly between the original and recreated versions of SUR database. One minor difference is that the original system, for example, displays only 10 items at a time. A more notable difference between the interfaces is how each system handles displaying associated item instances of multiple cardinality. Patients, for example, can have multiple Surgeries associated with them. The Patient profiles in the original system displays a list of each of the surgeries associated with the patient. The item profile interface for CELO, on the other hand, simply provides a link to a list of the surgeries associated with the patient. This same approach of providing a link to a list of associated items is taken in the original CSM database and is appropriate for certain sets of research data. The original SUR database's approach, however, may provide a friendlier browsing environment for other sets of data, and CELO is not currently able to recreate its interface design.



Single Unit Recording Database

sur_v3 version 3

Single Unit Recording Database

Recent Verbal Memory Study Home

Overview

Study of extracellular neuronal activity during memory tasks

Create or Find Items

Data Class	Description	Actions
Patient	Patient participating in the study	🔍 ✚ ✚

Data Views

Query name	Description
Browse Patients	List of study patients
Browse Events	List of events

Data Store Configuration

> Data Store Name and Description
> Data Classes
> Data Views
> Save as Template

Figure 27. SUR Database Comparison for Accessing Data

Query Electrode Neuron by Patient

Along with each patient profile in the original SUR database is a link for listing all electrodes associated with that patient. Determining this association requires following table relationships starting from the Patient, going to the patient's surgeries, continuing to each of these surgeries' trial protocols, and finally ending with the electrodes associated with the trial protocols. CELO does not support a method for including this type of query in each patient profile page. The view item interface of CELO by default only provides the attribute values of that item and a link to a list of items that are directly associated with it. Any distant associations implied by multi-table relationships can only be viewed by following the appropriate links to navigate down the object hierarchy. The query electrode by patient link is a unique feature of the original SUR database that CELO is currently unable to recreate.

5.4 Protein Interaction Study

Dr.Richard Morrison of the University of Washington's Department of Neurological Surgery performs research that focuses on neuronal damage due to injury or disease such as stroke, seizures, AIDS and neurogenerative diseases. One research study performed by the lab was a proteome analysis study of neuronal death using mass spectrometry technologies[70]. Dr. Morrison is now spearheading a research effort that will use methods similar to those used for the proteome analysis and that will require a collaboration of multiple laboratories. The goal of the protein interaction project is to integrate data collected from proteomics, genomics, and animal model studies performed in multiple labs, as well as information from public protein binding and publication databases. To help manage the integration of all these data, Dr. Morrison envisions a set of graphical tools that will allow researchers to visualize the strength of evidence for particular protein binding partners and to easily access related data. Before these tools can be created, the collaborating laboratories must first be able to store and organize their proteomics data into a database that can be easily accessed by other participating laboratories. We have used CELO to create a prototype system that will satisfy this informatics need. We describe how we used the customization tools to build the features of the system and discuss the benefits and drawbacks of the resulting product.

Defining Representations of Data

Unlike the Eye Lab, CSM studies, and SUR studies for which pre-existing data management systems were used as models to build a CELO-based system, the protein interaction study required a system to be designed and developed for the first time. To allow us to define the database representation of data, the lab researchers provided us with a sample spreadsheet of data that needed to be managed. The spreadsheet listed two example cell sample preparations along with a list of proteins likely to exist within the samples as determined by mass spectrometry analysis. Along with a list of alternate names for a protein, two links to the protein entries in public databases, AmiGO[71] and UniProt[72], were specified. A list of peptides identified by mass spectrometry analysis, along with parameters that indicate the certainty of the identification, is also associated with each protein for the sample.

Because we had not created a default CELO template for this new type of study being managed, we had to define the Data Classes from scratch. Naturally, we defined Data Classes to represent Sample, Protein and Peptide. After entering some items into the database, however, we realized that the research data would be better represented using a slightly different database schema. We therefore used the web tools to modify an existing Data Class and to add a new Data Class. Unfortunately, the modification resulted in a loss of data, requiring us to re-enter some data. The process proved to be inefficient, demonstrating the value of a schema evolution mechanism that can automatically move existing data between tables. This scenario emphasizes why our system requirement for features to facilitate evolution of data representations is so critical.

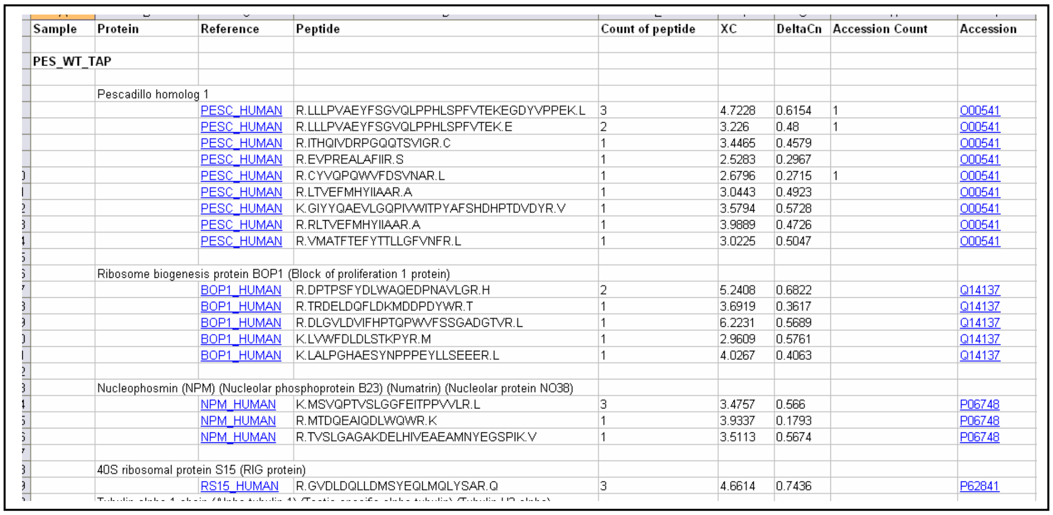
Once we were satisfied with the defined data representations for the Protein Interaction study, we saved the definitions as a CELO template. The schema reuse benefit of the template system is highlighted by the goals of the Protein Interaction study. The study involves a collaboration of laboratories, each of which will likely need to manage proteomics data similar to that recorded in the spreadsheet we were provided. These laboratories can now simply use the template that we created to represent their research data and will not suffer from the same mistakes that we did when defining the Data Classes from scratch.

Creating and Viewing Items

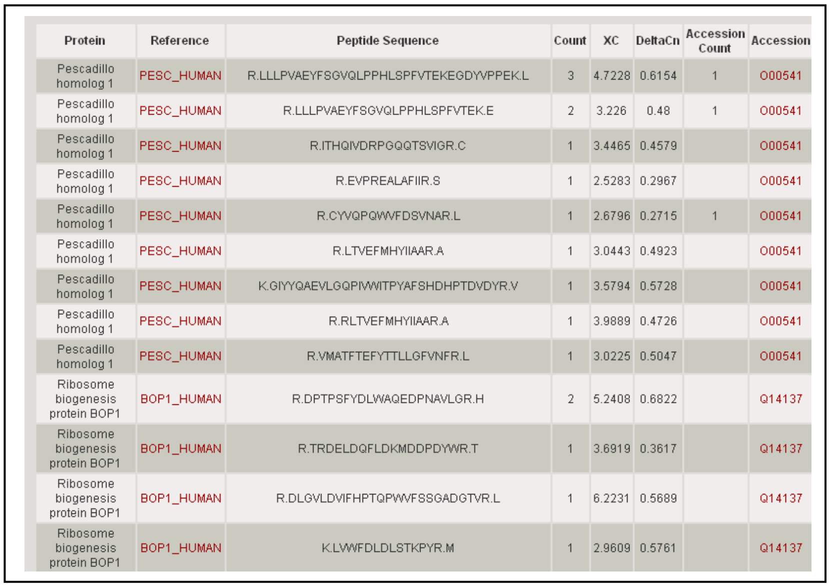
The user interfaces generated by CELO for creating items in the Protein Interaction database were easy to use and understand. The feature for entering multiple items on one page was useful for increasing the efficiency of entering data specified in the provided spreadsheet. Limitations of the interface for entering multiple items included the inability to use the arrow keys to navigate through the various input fields and the inability to copy and paste multiple cells from a spreadsheet to fill the values of multiple input fields. A feature to upload and parse a spreadsheet of data in order to automatically populate a database would also have been valuable to facilitate data entry in this scenario. Features similar to this were requested by the users of the Eye Lab image repository and the CSM database. This observation suggests that it may be a common need among laboratories to improve the efficiency of data entry for these database systems and to reduce the duplication of work already performed recording data into spreadsheets.

CELO’s predefined URL data type was a valuable feature for creating the Protein Interaction system. Two of the attributes for the Protein Data Class specified URL links to public protein databases. The user interface generated for creating new Protein items automatically displayed input fields relevant for the two URL type attributes. Users were provided two text boxes for specifying a URL item: one to enter a URL link label and the other to enter the actual URL web address. CELO then also performs special formatting to display URL attribute types such that when users view Protein items, the URL labels are displayed as links that can be followed to the public protein database websites.

The Protein Interaction data management system that we built also provides researchers with the ability to perform some useful queries. Users can, for example, list all the proteins identified in a sample, find all the samples that a particular protein was identified in, or display all the peptides identified in a sample. The nature of the protein interaction study data, however, reveals a weakness of the querying system. The results of a query are displayed in a rigid format and users have few options for customizing the formatting for their specific needs. The researchers of the protein interaction study, for example, used indentation within their spreadsheet in order to clearly express which proteins were identified in a sample and which peptides were associated with each of these proteins. The CELO query results that provide the same information, however, are displayed in a default format that makes it more difficult for users to quickly comprehend the data. The results table displayed without indentations, for example, less clearly delineates a set of peptides associated with one protein from a set of peptides associated with the next protein (Figure 28).



Sample	Protein	Reference	Peptide	Count of peptide	XC	DeltaCn	Accession Count	Accession
PES_WT_TAP								
	Pescadillo homolog 1							
		PESC_HUMAN	R.LLLPVAEYFSGVQLPPHLSPFVTEKEGDYVPPEK.L	3	4.7228	0.6154	1	O00541
		PESC_HUMAN	R.LLLPVAEYFSGVQLPPHLSPFVTEK.E	2	3.226	0.48	1	O00541
		PESC_HUMAN	R.ITHQIVDRPGQQTSVIGR.C	1	3.4465	0.4579		O00541
		PESC_HUMAN	R.EVPREALAFIIR.S	1	2.5283	0.2967		O00541
		PESC_HUMAN	R.CYVQPQWVFDSVNAR.L	1	2.6796	0.2715	1	O00541
		PESC_HUMAN	R.LTVEFMHYIIAAR.A	1	3.0443	0.4923		O00541
		PESC_HUMAN	K.GIYYQAEVLGQPIVWITPYAFSHDHPTDVDYR.V	1	3.5794	0.5728		O00541
		PESC_HUMAN	R.RLTVEFMHYIIAAR.A	1	3.9889	0.4726		O00541
		PESC_HUMAN	R.VMATFTEFYTTLLGFVNFR.L	1	3.0225	0.5047		O00541
	Ribosome biogenesis protein BOP1 (Block of proliferation 1 protein)							
		BOP1_HUMAN	R.DPTPSFYDLWAQEDPNAVLGR.H	2	5.2408	0.6822		Q14137
		BOP1_HUMAN	R.TRDELDQFLDKMDDPDYWR.T	1	3.6919	0.3617		Q14137
		BOP1_HUMAN	R.DLGVLDVIFHPTQPWVFSSGADGTVR.L	1	6.2231	0.5689		Q14137
		BOP1_HUMAN	K.LVWFDLDLSTKPYR.M	1	2.9609	0.5761		Q14137
		BOP1_HUMAN	K.LALPGHAESYNPPPEYLLSEEER.L	1	4.0267	0.4063		Q14137
	Nucleophosmin (NPM) (Nucleolar phosphoprotein B23) (Numatrin) (Nucleolar protein NO38)							
		NPM_HUMAN	K.MSVQPTVSLGGFEITPPVVLR.L	3	3.4757	0.566		P06748
		NPM_HUMAN	R.MTDQEAIQDLWQWR.K	1	3.9337	0.1793		P06748
		NPM_HUMAN	R.TVSLGAGAKDELHIVEAEAMNYEGSPIK.V	1	3.5113	0.5674		P06748
	40S ribosomal protein S15 (RIG protein)							
		RS15_HUMAN	R.GVDLDQLLDMSYEQLMQLYSAR.Q	3	4.6614	0.7436		P62841



Protein	Reference	Peptide Sequence	Count	XC	DeltaCn	Accession Count	Accession
Pescadillo homolog 1	PESC_HUMAN	R.LLLPVAEYFSGVQLPPHLSPFVTEKEGDYVPPEK.L	3	4.7228	0.6154	1	O00541
Pescadillo homolog 1	PESC_HUMAN	R.LLLPVAEYFSGVQLPPHLSPFVTEK.E	2	3.226	0.48	1	O00541
Pescadillo homolog 1	PESC_HUMAN	R.ITHQIVDRPGQQTSVIGR.C	1	3.4465	0.4579		O00541
Pescadillo homolog 1	PESC_HUMAN	R.EVPREALAFIIR.S	1	2.5283	0.2967		O00541
Pescadillo homolog 1	PESC_HUMAN	R.CYVQPQWVFDSVNAR.L	1	2.6796	0.2715	1	O00541
Pescadillo homolog 1	PESC_HUMAN	R.LTVEFMHYIIAAR.A	1	3.0443	0.4923		O00541
Pescadillo homolog 1	PESC_HUMAN	K.GIYYQAEVLGQPIVWITPYAFSHDHPTDVDYR.V	1	3.5794	0.5728		O00541
Pescadillo homolog 1	PESC_HUMAN	R.RLTVEFMHYIIAAR.A	1	3.9889	0.4726		O00541
Pescadillo homolog 1	PESC_HUMAN	R.VMATFTEFYTTLLGFVNFR.L	1	3.0225	0.5047		O00541
Ribosome biogenesis protein BOP1	BOP1_HUMAN	R.DPTPSFYDLWAQEDPNAVLGR.H	2	5.2408	0.6822		Q14137
Ribosome biogenesis protein BOP1	BOP1_HUMAN	R.TRDELDQFLDKMDDPDYWR.T	1	3.6919	0.3617		Q14137
Ribosome biogenesis protein BOP1	BOP1_HUMAN	R.DLGVLDVIFHPTQPWVFSSGADGTVR.L	1	6.2231	0.5689		Q14137
Ribosome biogenesis protein BOP1	BOP1_HUMAN	K.LVWFDLDLSTKPYR.M	1	2.9609	0.5761		Q14137

Figure 28. Protein Interaction Study Data Formatting Comparison

5.5 Future Evaluation Plans

The CELO based data management systems built for the Eye Lab, Cortical Stimulation Mapping (CSM), Single Unit Recording (SUR), and Protein Interaction studies demonstrate the range of informatics needs that CELO is able to satisfy. These four examples, however, still do not sufficiently represent a wide range of biomedical research areas to validate the generalizability of the system. We have also not yet directly evaluated the effectiveness of CELO's distributed resource model approach or whether the system has effectively provided tools for researchers to build their own data management systems. Another critical aspect of software system evaluation we plan to perform is a formal study of end user satisfaction.

Our future evaluation plans involve introducing the system to additional laboratories in need of informatics solutions. The greater number of laboratories the system is tested in, the better we will be able to evaluate its generalizability and to identify new features that will improve its generalizability. Instead of using CELO to create systems for these additional laboratories ourselves, we will ask the researchers to build their own systems. A formal evaluation will utilize surveys and interviews to assess the system's usability, its ability to meet the researchers' needs, and a general sense of end user satisfaction. A potential issue with CELO's distributed resource model is whether laboratories feel secure about storing data on a remote server that is shared with other laboratories. Another concern with the resource sharing design is how well the server can handle high volume simultaneous access. The trade-offs between the cost of laboratory data management systems and the performance of these systems should be investigated. Our future evaluations must address these issues.

5.6 Summary of Evaluation Results

We evaluated CELO by using its web-based tools to recreate three existing data management systems we have previously built and to create a new system for a laboratory we had not previously worked with. The evaluation demonstrated that our hypothesis was satisfied relatively well, with CELO able to implement various features of data management systems very efficiently. The evaluation, however, also revealed that CELO was not able to meet all of the informatics needs of laboratories. As might be expected, many of the more complex or unique features of the original systems could not be replicated.

The evaluation also showed that CELO was able to recreate some of the existing laboratory data management systems more completely than others. Note in the "Replicated" column of Table 4 that most of the major features of The Eye Lab Image Repository, SUR database and Protein Interaction study could be completely implemented using CELO. All of the major features of the CSM database, on the other hand, could only be partially implemented. For example, although an interface for viewing Patient items was efficiently implemented using CELO, the exact formatting used for the original CSM database could not be replicated. Also note that three of the systems used for the evaluation contained at least one unique feature that could not be recreated by CELO at all. For example, the automatically generated image matrices of the Eye Lab Image Repository and support for the Visual Brain Mapper application of the CSM database could not be implemented by CELO. This evaluation result demonstrates the importance of supporting the system requirement for plugins for custom features.

A major strength of the CELO system demonstrated by the evaluation is the system's ability to implement features efficiently. The "Implementation Time" column of Table 4 demonstrates that nearly all the major features of the original systems that could be recreated by CELO were very efficiently developed. This suggests that laboratory data management systems can be quickly built and customized using the CELO system tools.

Table 4. Summary of Evaluation Results: Implementing Major Features

Feature	Replicated	Implementation Time
Eye Lab Image Repository		
Controlling User Access	++	++
Organizing Data from Separate Experiments	++	++
Defining Representations of Data	++	++
Creating and Viewing Items	++	++
Finding Images	++	++
Creating Image Matrices for Image Comparison and Analysis		
The Cortical Stimulation Mapping Database		
Controlling User Access	+	++
Defining Representations of Data	+	++
Creating and Viewing Items	+	++
Access to Data	+	+
System Navigation	+	++
Visual Brain Mapper Application		
Single Unit Recording Database		
Defining Representations of Data	++	++
Access to Data	++	++
Creating and Viewing Items	+	++
Query Electrode Neuron by Patient		
Protein Interaction Study		
Defining Representations of Data	++	+
Creating and Viewing Items	+	++

Our evaluation also assessed the importance of each of the nine system requirements for each of the four laboratories that we created a CELO system for. This assessment helps to further validate our selection of requirements, as well as suggest whether CELO is an appropriate informatics solution for each of the laboratories of our evaluation. A summary of our findings is outlined in Table 5. This table reintroduces the matrix reviewing the existing informatics solutions based on the nine system requirements with the addition of indicators of the importance of each of these requirements for the four systems developed for our evaluation. For each of the four systems, we have indicated the three requirements that we believe are most critical for the needs of the laboratory. The matrix helps indicate whether CELO is an appropriate informatics solution for each lab or if another solution might better serve the laboratory's needs. For example, because CELO does not currently support plugins, the Eye

Lab may find another informatics solution such as WIRM or Microsoft Access to be a better choice for creating a data management system.

Table 5. Summary of Evaluation Results: Satisfying System Requirements

	Inexpensive	Short Development Time	Customizable graphical user interfaces	Features facilitating database design	Support for diverse data types	Powerful querying capabilities	Support for sharing over the internet	Plugins for customizations	Evolution of Data Representation
NeuroSys	X	X	X	X			X		
SenseLab	X	X		X			X		
Microsoft Access		X	X	X	X	X		X	
WIRM	X				X	X	X	X	
Commercial LIMS					X	X	X	X	
CELO	X	X	X	X	X	X	X		
The Eye Lab	+	+	+	+	++	++	+	++	+
CSM Database	+	+	++	+	+	+	++	++	+
SUR Database	+	+	++	++	++	+	+	+	+
Protein Interaction	+	+	+	+	++	+	++	+	++

Chapter 6: Discussion and Conclusions

We built CELO acknowledging that the system would not be able to satisfy the informatics needs of every laboratory. Our evaluation has confirmed that the system is not a one-size-fits-all solution. The attempt to recreate the Cortical Stimulation Mapping (CSM) database, in particular, provides an example in which a CELO based system is not the ideal solution. Although CELO configuration tools were able to generate the database schema, a couple of useful queries, and some effective interfaces for entering data, it was unable to capture the most useful features of the original system. Examples of such features of the CSM database include predefined data views generated by multiple complex queries, custom experiment workflow tools, and integration of a domain specific application. Most of the custom scripts of the original system would also be required for the CELO based system in order to satisfy the laboratory's needs. The Eye Lab, Single Unit Recording (SUR) study, and Protein Interaction study required more basic data management needs such as simple data organization, visualization and retrieval functions that a CELO based system was more successfully able to fulfill.

A top priority in our plans for future work includes implementing features to satisfy the two system requirements that CELO does not currently support: plugins for customization and evolution of data representation. The significance of these requirements was only emphasized by the results of our evaluation. Even the laboratories with mostly basic data management needs also required some custom functionality specific to a research area. For example, the Eye Lab needed a feature for creating image matrices in order to compare and analyze cataract formation in mice, and the SUR study needed each Patient view to include a specialized query for listing the electrodes associated with the Patient. If CELO supported plugins, customizations such as these could be integrated into a laboratory's data management system to satisfy unique needs. Laboratories can already drop custom written scripts into their dedicated filesystem directory on the CELO server such that the scripts can be accessed by a web browser. In order to fully support plugins, we plan to add a mechanism to link to these custom scripts from the generic web interface. Our idea for implementing this mechanism is to create a new table in each lab database that specifies the scripts written to satisfy a lab's unique needs. The generic CELO CGI scripts would then be modified to query this table to display the appropriate links for executing the custom scripts.

Evaluation of the Protein Interaction system also illustrated the importance of the schema evolution requirement. During creation of the Protein Interaction system, we were required to evolve the database schema in a manner that caused data to be lost that had to be re-entered. A mechanism to efficiently evolve the database schema without data loss would have been very valuable in this scenario. For example, a web interface that allows users to easily specify data to copy from one database table to another would have eliminated the need to re-enter data in our scenario, and is one feature that would help satisfy the schema evolution requirement.

Although our evaluations have demonstrated how CELO has been designed to effectively meet the remaining seven requirements, they have also illustrated how the system might be improved to better meet the requirements and to better serve the needs of laboratories. The customizable user interfaces requirement, for example, led to the development of configuration tools that allow the users to specify labels for database table and column names, determine the order in which attribute are displayed, and select widgets for creating and querying data. Our evaluation has shown that the user interface configuration options are not sufficient to meet all of the needs of research labs. For example, CELO was unable to create a user interface for the Protein Interaction Study that displayed query results in specific format. Queries constructed in CELO accurately returned a list of results for a query of samples, proteins, and peptides, but the formatting of the results made it difficult to distinguish which peptides are associated with which proteins, and which proteins are associated with which samples. Formatting the results with indentations, as was done by the researchers in their spreadsheet, would more clearly make these distinctions. Adding user interface configuration options such that researchers could specify formatting options such as this is an area of future work for the CELO system.

We developed the CELO template system in order to help labs design databases for representing research data, and our evaluation has demonstrated how the templates help make this task easier and more efficient. Many laboratories, however, will not find a template that fits their needs. In the early stages of CELO use, before more templates are created, laboratories will often still need to define their own schemas from scratch. This task can be quite complex, as highlighted by the need to evolve the schema for the Protein Interaction study database. Tools such as the visualization tools available in Microsoft Access may help improve CELO's ability to facilitate database design and may be worth adding to future versions of CELO.

CELO's querying features allowed us to create some very valuable views of data, including lists of data joining multiple tables. Yet our evaluation showed that some data views, such as the patient status list valuable for the CSM study, could not be created using a single SQL query. Future work on the CELO system may include developing features for constructing more complex views of data using multiple queries. Another issue with CELO's querying features is that its advanced query method, which allows users to construct SQL statements, requires the user to have a SQL programming background. Researchers may not have the time or desire to learn SQL. Developing additional querying tools that allow users to construct complex queries without SQL knowledge is another potential area for future work.

CELO has also demonstrated its support for diverse data types. The File and URL data types predefined within the CELO system has proved to be very valuable. CELO's special processing of File types enabled the Eye Lab members to use their system to easily organize and retrieve several mouse eye images for analysis. The Protein Interaction system utilizes the URL type to associate proteins identified in their experimental studies with links to web pages displaying protein annotations from public protein databases. We acknowledge that there are other data types common to biomedical research that would benefit from special processing. Future work on CELO includes researching potential data types to predefine, such as types representing data relevant to genomics or proteomics studies. There already exist some research efforts that explore various models for representing these complex data[6, 12, 73-75].

The Protein Interaction study is an example of a research effort that requires multi-laboratory collaboration and integration of information from public biomedical databases. Because CELO is a web-based system, it allows laboratories to easily share research data with other remote labs. We expect several collaborations, including the Protein Interaction study, to also require the integration of data from multiple laboratory databases as well as from public biological databases. We therefore plan to research different methods for integrating data from multiple databases. The need for these biological data integration tools is well known, and several research efforts have already been launched in this area[76-78]. Our future work includes exploring these options as potential mechanisms for laboratories to integrate data from their CELO-based systems with other laboratory databases and publicly available databases.

All of the potential areas of future work, from additional user interface configuration options to integration of multiple biomedical databases, could increase the value of the CELO system. An issue with adding more features, however, is that it can also result in a higher the learning curve for using the system. We will use caution when selecting the new features to implement in the system, considering the trade-off between usability and customizability. CELO, in its current state, has already demonstrated its promise as a valuable tool for labs that need an inexpensive and quick solution for basic data management needs. We believe that many biomedical laboratories fall within this category and that CELO, therefore, has the potential to be adopted by the biomedical research community. One challenge we must face, however, is identifying labs that are willing to spend the time and effort required to use CELO in practice. Our experiences thus far have indicated that although labs recognize a need for data management solutions, many are also weary of spending time investigating potential solutions. As more labs recognize the benefits of CELO, we hope to further validate the system as a valuable tool for research labs over a wide range of research domains and with a diversity of informatics needs.

Bibliography

1. Jakobovits RM, Rosse C, Brinkley JF. WIRM: an open source toolkit for building biomedical web applications. *J Am Med Inform Assoc.* Nov-Dec 2002;9(6):557-570.

2. Pittendrigh S, Jacobs G. NeuroSys: a semistructured laboratory database. *Neuroinformatics.* 2003;1(2):167-176.

3. Marenco L, Tosches N, Crasto C, Shepherd G, Miller PL, Nadkarni PM. Achieving evolvable Web-database bioscience applications using the EAV/CR framework: recent advances. *J Am Med Inform Assoc.* Sep-Oct 2003;10(5):444-453.

4. Imbert MC, Nguyen VK, Granjeaud S, Nguyen C, Jordan BR. 'LABNOTE', a laboratory notebook system designed for academic genomics groups. *Nucleic Acids Res.* Jan 15 1999;27(2):601-607.

5. Hong P, Wong WH. GeneNotes--a novel information management software for biologists. *BMC Bioinformatics.* Feb 1 2005;6(1):20.

6. Goh CS, Lan N, Echols N, et al. SPINE 2: a system for collaborative structural proteomics within a federated database framework. *Nucleic Acids Res.* Jun 1 2003;31(11):2833-2838.

7. Sanchez-Villeda H, Schroeder S, Polacco M, et al. Development of an integrated laboratory information management system for the maize mapping project. *Bioinformatics.* Nov 1 2003;19(16):2022-2030.

8. Rubin DL, Shafa F, Oliver DE, Hewett M, Altman RB. Representing genetic sequence data for pharmacogenomics: an evolutionary approach using ontological and relational models. *Bioinformatics.* 2002;18 Suppl 1:S207-215.

9. Lander ES, Linton LM, Birren B, et al. Initial sequencing and analysis of the human genome. *Nature.* Feb 15 2001;409(6822):860-921.

10. Seeberger TM, Matsumoto Y, Alizadeh A, Fitzgerald PG, Clark JI. Digital image capture and quantification of subtle lens opacities in rodents. *J Biomed Opt.* Jan-Feb 2004;9(1):116-120.

11. Insel TR, Volkow ND, Li TK, Battey JF, Jr., Landis SC. Neuroscience networks: data-sharing in an information age. *PLoS Biol.* Oct 2003;1(1):E17.

12. Goodman N, Rozen S, Stein LD, Smith AG. The LabBase system for data management in large scale biology research laboratories. *Bioinformatics.* 1998;14(7):562-574.

13. Graves M. Data note system for capturing laboratory data. *Genomics.* Jul 15 1997;43(2):232-236.

14. Fellmann D, Pulokas J, Milligan RA, Carragher B, Potter CS. A relational database for cryoEM: experience at one year and 50 000 images. *J Struct Biol.* Mar 2002;137(3):273-282.

15. Metoz F, Sherman MB, Schmid MF. Adopting a database as a solution to managing electron image data. *J Struct Biol.* Feb-Mar 2001;133(2-3):170-175.

16. Morris C, Wood P, Griffiths SL, Wilson KS, Ashton AW. MOLE: a data management application based on a protein production data model. *Proteins.* Feb 1 2005;58(2):285-289.

17. Nayler O, Stamm S. ScienceLabDatabase: a computer program to organize a molecular biology laboratory. *Biotechniques.* Jun 1999;26(6):1186-1191.

18. Navarro JD, Talreja N, Peri S, et al. BioBuilder as a database development and functional annotation platform for proteins. *BMC Bioinformatics.* Apr 20 2004;5(1):43.

19. Hinshaw KP, Poliakov AV, Moore EB, Martin RF, Shapiro LG, Brinkley JF. Shape-based cortical surface segmentation for visualization brain mapping. *Neuroimage.* Jun 2002;16(2):295-316.

20. Jakobovits RM, Brinkley JF. Managing medical research data with a Web-Interfacing Repository Manager. *Proc AMIA Annu Fall Symp.* 1997:454-458.

21. Marshall WW, Haley RW. Use of a secure Internet Web site for collaborative medical research. *Jama.* Oct 11 2000;284(14):1843-1849.

22. Fong C, Rosse C, Clark J, Shapiro L, Brinkley J. An Ontology-based Image Repository for a Biomedical Research Lab. *Medinfo.* 2004;2004(2004(CD)):1598.

23. The Eye Lab Image Repository. http://eyelab.biostr.washington.edu/.

24. Marenco L, Nadkarni P, Skoufos E, Shepherd G, Miller P. Neuronal database integration: the Senselab EAV data model. *Proc AMIA Symp.* 1999:102-106.

25. Microsoft Access. http://office.microsoft.com/en-us/FX010857911033.aspx.

26. StarLIMS. http://www.starlims.com/.

27. LabVantage. http://www.labvantage.com/.

28. LabWare. http://www.labware.com/.

29. NeuroSys. http://neurosys.cns.montana.edu.

30. SenseLab. http://senselab.med.yale.edu/senselab/.

31. Miller PL, Nadkarni P, Singer M, Marenco L, Hines M, Shepherd G. Integration of multidisciplinary sensory data: a pilot model of the human brain project approach. *J Am Med Inform Assoc.* Jan-Feb 2001;8(1):34-48.

32. Nadkarni PM, Marenco L, Chen R, Skoufos E, Shepherd G, Miller P. Organization of heterogeneous scientific data using the EAV/CR representation. *J Am Med Inform Assoc.* Nov-Dec 1999;6(6):478-493.

33. Microsoft SQL Server. http://www.microsoft.com/sql/default.mspx.

34. Prague CN. *Access 2003 Bible*. Indianapolis, Ind: Wiley Publications; 2004.

35. Web Interfacing Repository Manager. http://www.wirm.org/.

36. Jakobovits RM. *The Web interfacing repository manager : a framework for developing web-based experiment management systems* [Ph. D.]. Seattle: Computer Science and Engineering, University of Washington; 1999.

37. Jakobovits RM, Modayur B, Brinkley JF. A Web-based repository manager for brain mapping data. *Proc AMIA Annu Fall Symp.* 1996:309-313.

38. Avery G, McGee C, Falk S. Implementing LIMS: a "how-to" guide. *Anal Chem.* Jan 1 2000;72(1):57A-62A.

39. Jakobovits R, Soderland SG, Taira RK, Brinkley JF. Requirements of a Web-based experiment management system. *Proc AMIA Symp.* 2000:374-378.

40. cancer Laboratory Management System. http://calims.nci.nih.gov/caLIMS/.

41. Flow Cytometry Laboratory Management System. http://bioinformatics.fccc.edu/software/OpenSource/flowLIMS/flowlims.shtml.

42. GnosisLIMS. http://www.gnosislims.org/wiki/tiki-index.php.

43. Patel VL, Kushniruk AW. Interface design for health care environments: the role of cognitive science. *Proc AMIA Symp.* 1998:29-37.

44. Olson GM, Olson JS. Human-computer interaction: psychological aspects of the human use of computing. *Annu Rev Psychol.* 2003;54:491-516.

45. Bellgard MI, Hiew HL, Hunter A, Wiebrands M. ORBIT: an integrated environment for user-customized bioinformatics tools. *Bioinformatics.* Oct 1999;15(10):847-851.

46. Jacques PF, Taylor A, Moeller S, et al. Long-term nutrient intake and 5-year change in nuclear lens opacities. *Arch Ophthalmol.* Apr 2005;123(4):517-526.

47. van Wijngaarden P, Coster DJ, Brereton HM, Gibbins IL, Williams KA. Strain-dependent differences in oxygen-induced retinopathy in the inbred rat. *Invest Ophthalmol Vis Sci.* Apr 2005;46(4):1445-1452.

48. Alizadeh A, Clark J, Seeberger T, Hess J, Blankenship T, FitzGerald PG. Characterization of a mutation in the lens-specific CP49 in the 129 strain of mouse. *Invest Ophthalmol Vis Sci.* Mar 2004;45(3):884-891.

49. Clark JI, Livesey JC, Steele JE. Delay or inhibition of rat lens opacification using pantethine and WR-77913. *Exp Eye Res.* Jan 1996;62(1):75-84.

50. Brodie R, Smith AJ, Roper RL, Tcherepanov V, Upton C. Base-By-Base: single nucleotide-level analysis of whole viral genome alignments. *BMC Bioinformatics.* Jul 14 2004;5(1):96.

51. PubMed. http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?DB=pubmed.

52. Entrez Nucleotide sequence database (GenBank). http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Nucleotide.

53. Wilkinson MD, Block D, Crosby WL. Genquire: genome annotation browser/editor. *Bioinformatics.* Oct 2002;18(10):1398-1399.

54. CGI - Common Gateway Interface. http://www.w3.org/CGI/.

55. W3C HyperText Markup Language (HTML) Home Page. http://www.w3.org/MarkUp/.

56. Wall L, Christiansen T, Schwartz RL. *Programming Perl*. 2nd ed ed. Sebastopol, CA: O'Reilly & Associates; 1996.

57. Comprehensive Perl Archive Network (CPAN). http://www.cpan.org/.

58. Stein L. CGI.pm http://search.cpan.org/dist/CGI.pm/.

59. Brian D. SimpleObject. http://search.cpan.org/~dbrian/XML-SimpleObject-0.53/SimpleObject.pm.

60. MySQL. http://dev.mysql.com/.

61. W3C Extensible Markup Language (XML). http://www.w3.org/XML/.

62. W3C Cascading Style Sheets Home Page. http://www.w3.org/Style/CSS/.

63. Ojemann GA. The neurobiology of language and verbal memory: observations from awake neurosurgery. *Int J Psychophysiol.* May 2003;48(2):141-146.

64. Lucas TH, 2nd, McKhann GM, 2nd, Ojemann GA. Functional separation of languages in the bilingual brain: a comparison of electrical stimulation language mapping in 25 bilingual patients and 117 monolingual control patients. *J Neurosurg.* Sep 2004;101(3):449-457.

65. Corina DP, Gibson EK, Martin R, Poliakov A, Brinkley J, Ojemann GA. Dissociation of action and object naming: evidence from cortical stimulation mapping. *Hum Brain Mapp.* Jan 2005;24(1):1-10.

66. Ojemann GA, Schoenfield-McNeill J, Corina DP. Anatomic subdivisions in human temporal cortical neuronal activity related to recent verbal memory. *Nat Neurosci.* Jan 2002;5(1):64-71.

67. Lucas TH, Jr., Schoenfield-McNeill J, Weber PB, Ojemann GA. A direct measure of human lateral temporal lobe neurons responsive to face matching. *Brain Res Cogn Brain Res.* Dec 2003;18(1):15-25.

68. Ojemann GA, Schoenfield-McNeill J, Corina D. Different neurons in different regions of human temporal lobe distinguish correct from incorrect identification or memory. *Neuropsychologia.* 2004;42(10):1383-1393.

69. Li H, Brinkley J, Gennari J. Semi-automatic Database Design for Neuroscience Experiment Management Systems. *Medinfo.* 2004;2004:1717.

70. Johnson MD, Yu LR, Conrads TP, et al. Proteome analysis of DNA damage-induced neuronal death using high throughput mass spectrometry. *J Biol Chem.* Jun 18 2004;279(25):26685-26697.

71. AmiGO. http://www.godatabase.org.

72. UniProt. http://www.pir.uniprot.org/.

73. Pajon A, Ionides J, Diprose J, et al. Design of a data model for developing laboratory information management and analysis systems for protein production. *Proteins.* Feb 1 2005;58(2):278-284.

74. Taylor CF, Paton NW, Garwood KL, et al. A systematic approach to modeling, capturing, and disseminating proteomics experimental data. *Nat Biotechnol.* Mar 2003;21(3):247-254.

75. Raymond S, O'Toole N, Cygler M. A data management system for structural genomics. *Proteome Sci.* Jun 21 2004;2(1):4.

76. Shaker R, Mork P, Brockenbrough JS, Donelson L, Tarczy-Hornoch P. The BioMediator System as a Tool for Integrating Biologic Databases on the Web. *Proceedings of the Workshop on Information Integration on the Web.* August 2004 2004.

77. Hubbard T, Andrews D, Caccamo M, et al. Ensembl 2005. *Nucleic Acids Res.* Jan 1 2005;33(Database issue):D447-453.

78. Chung SY, Wong L. Kleisli: a new tool for data integration in biology. *Trends Biotechnol.* Sep 1999;17(9):351-355.