**Distribution Agreement**

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

_____ _____

Serdar Kurtkaya Date

Designing a Laboratory Information Management System by Using Open-Source Tools

OSIRIS: Open Source Integrated Research Information System

By

Serdar Kurtkaya

Master of Science in Public Health

Department of Biostatistics and Bioinformatics

_____

Vicki S. Hertzberg

Advisor

_____

Lance A. Waller

Department Chair

# Designing a Laboratory Information Management System by Using Open-Source Tools

## OSIRIS: Open Source Integrated Research Information System

By

Serdar Kurtkaya

B.S., Bogazici University, 2001

M.S., Emory University, 2005

Advisor: Vicki S. Hertzberg, PhD

An abstract of

a Special Studies Report submitted to the

Public Health Informatics Program

Rollins School of Public Health of Emory University

in partial fulfillment of the requirements for the degree of

Master of Science in Public Health

in Department of Biostatistics and Bioinformatics

2012

# Abstract

Designing a Laboratory Information Management System by Using Open-Source Tools

OSIRIS: Open Source Integrated Research Information System

By Serdar Kurtkaya

Storing all scientific data in a small-scale drug discovery lab requires comprehensive approach. In order to tackle this problem, a centralized resource for cross-disciplinary interaction called Open Source Integrated Research Information System (OSIRIS) has been developed to create a compound registration and project management environment. OSIRIS has been implemented as a group-wide accessible webpage to avoid individual user-side installation and platform compatibility problems.

This system includes modules for creation of projects with custom user privileges, a document management system based on projects, registration of new chemical compounds, text and structure based search, addition of experimental results to individual compounds, structure and data export, administrator controlled ticket management, and is PC/Mac browser platform independent.

OSIRIS has been created by using open-source tools to keep the cost to minimum and to give maximum customization of the capabilities of the system. It can be modified to work with different public health informatics related projects; such as, creation of hazardous chemicals database for environmental health research, or a collaboration platform for teams working in different parts of the world.

This work has been done at the Emory Institute for Drug Discovery (EIDD) in Atlanta, GA over a 2 year period, and is based on user feedback. The source code will be available free of charge to academic and non-profit institutions in the spirit of open-source collaboration.

# Designing a Laboratory Information Management System by Using Open-Source Tools

## OSIRIS: Open Source Integrated Research Information System

By

Serdar Kurtkaya

B. S., Bogazici University, 2001

M.S., Emory University, 2005

Advisor: Vicki S. Hertzberg, PhD

A Special Studies Report submitted to the

Public Health Informatics Program

Rollins School of Public Health of Emory University

in partial fulfillment of the requirements for the degree of

Master of Science in Public Health

in Department of Biostatistics and Bioinformatics

2012

**TABLE OF CONTENTS**

**LIST OF FIGURES:**



**LIST OF GRAPHS:**



**LIST OF TABLES:**

**LIST OF ABBREVIATIONS:**

BCDMPK – Bioanalytical Chemistry, Drug Metabolism and Pharmacokinetics

CDX – a ChemDraw Program File

$EC_{50}$ – Half Maximal Effective Concentration

EIDD – Emory Institute for Drug Discovery

ER – Entity-Relationship

FK – Foreign Key

GB – Gigabyte

GHz – Gigahertz

ID – Identification Number

IT – Information Technology

LIMS – Laboratory Information Management Systems

MOL – a Molecular Representation File Format

MS – Microsoft

OSIRIS – Open Source Integrated Research Information System

PK – Primary Key

RAM – Random-Access Memory

SCSI – Small Computer System Interface

SD – Structure-Data

# 1 Chapter 1: Introduction

## *1.1 Background*

Emory Institute for Drug Discovery (EIDD)[1] is a research organization that focuses on public health problems by specializing in discovery of novel drugs. At the time of this report there were 30 EIDD members, which included synthetic chemists, biologists, pharmacologists, and information modelers.

EIDD scientists generate experimental data; such as, biological activity and human safety data, as well as report files that need to be cataloged. It is very important to keep a clean data organization within the Institute to review all the data required to make good drug discovery decisions.

The Information Technology (IT) group has been responsible for organizing the data flow between members. It is in the IT group's interest to use open-source tools to create the desired IT product. Open-source tools are preferred because of their custom framework, as well as, their low-cost overhead.

## *1.2 Problem Statement*

The Emory Institute for Drug Discovery has assembled the full complement of capabilities and expertise to collaborate with academic researchers at an early stage of the drug discovery process, and to progress emerging drug candidates toward regulatory approval and entry into human clinical evaluation. The Institute works with multiple collaborators and carries out multiple independent projects simultaneously.

In 2009, the EIDD recognized a need to have its own compound registration and document management system as its portfolio of collaborative projects grew and data management and communication became increasingly cumbersome. In order to solve this problem, an open-source tool named OSIRIS (Open Source Integrated Research Information System) was proposed to be created.

## *1.3 Purpose Statement*

In order to identify the exact purpose and scope of this new program all stakeholders (administrative staff, accountants, scientists, and programmers) came together for multiple meetings and interviews in the spring of 2009; and as a result, six core requirements were identified as critical to the final solution. The system will:

i. Support multiple projects with custom user access privileges

ii. Allow users to register new compounds

iii. Allow file-sharing at the project level

iv. Allow addition of experimental data to individual compounds

v. Be easy to use

vi. Be affordable

OSIRIS has been developed over two years based on user feedback. Its proven success is intimately linked to the latter and its customizable framework. The system has been implemented as a web-accessible and user-friendly tool that allows the scientists to avoid software installation and operating system compatibility issues.

The system includes modules for compound registration, document management, and a ticket system for experimental requests. Additionally, OSIRIS has been created by using open source resources for group-focused customization and economic considerations.

## *1.4 Significance Statement*

An essential component of a successful research organization is the ability to access any data that they have produced. A new data platform that was built by using open-source components would be a great tool for chemical data registration and data sharing. This would particularly be useful in environmental health area where chemicals are one of the main research areas.

The open-source nature of the program would encourage world-wide collaboration, and would make it very customizable and cost-effective for each organization’s needs.

# 2 Chapter 2: Review of Laboratory Information Management Systems

## *2.1 Commercial Laboratory Information Management Systems*

Commercial LIMS tools have been available since 1980s. There are highly automated LIMS tools such as ChemBioOffice Enterprise Suite[2] that can cover most of the core requirements. As in most commercial solutions, the price is the main sticking point to prevent using those software programs.

## *2.2 Open-Source Laboratory Information Management Systems*

Several open-source Laboratory Information Management Systems (LIMS) are available including Open-LIMS[3] and OpenELIS.[4]

Open-LIMS is an open-source Laboratory Information System that is very easy to install and use and comes with a strong community support group. Similarly, OpenELIS is an open-source Enterprise Laboratory Information System. It has a friendly user-interface; however, it lacks documentation of how to use it efficiently.

## *2.3 Existing Software vs. Core Requirements*

A comparison of commercial and open-source LIMS tools has been summarized in Table 1. Since none of these packages satisfied all core requirements, the decision was

made to develop an in-house solution. The outcome, an intranet website called OSIRIS, has been developed to create a compound registration and project management environment.

| Core Requirements | ChemBioOffice | Open-LIMS | OpenELIS |
|---|---|---|---|
| Support multiple projects with custom user access privileges | ✗ | ✓ | ✓ |
| Allow users to register new compounds | ✓ | ✗ | ✗ |
| Allow file-sharing at the project level | ✗ | ✓ | ✓ |
| Allow addition of experimental data to individual compounds | ✓ | ✗ | ✗ |
| Easy to use | ✓ | ✓ | ✗ |
| Affordable | ✗ | ✓ | ✓ |

**Table 1. Core Requirements vs. LIMS tools**

# 3 Chapter 3: Methodology

## *3.1 System Requirements*

The OSIRIS server can be installed on a system that supports the required services, i.e., WEBrick 1.3.1 webserver,[5] PostgreSQL 8.3,[6] Ruby 1.8.6,[7] Rails 1.2.6,[7] and pgchem::tigress 1.0[8] (Figure 1). Client-side requirements include a web browser and JChemPaint 3.1.2 installation for structure viewing.[9] All these programs listed are open-source and can be downloaded for free.

The OSIRIS server used for this publication was installed on a Dell Server PE840 with quad 2.40 GHz Intel Xeon X3220 processors, 4.00 GB of RAM and 500GB SCSI hard disk. The operating system for this machine is MS Windows Server 2008.

This stack of programs and server were selected due to:

- PostgreSQL: This database provided an open-source chemistry plug-in (pgchem::tigress) that was needed
- JChemPaint: A very simple and free chemistry viewing tool
- Ruby-on-Rails: It is a very structured framework that can handle various user and project privileges. Other tools such as PHP[10] or Perl[11] weren't used due to close integration of Ruby-on-Rails and pgchem:tigress chemistry plug-in
- WEBrick webserver: This webserver is the default webserver that comes with Ruby-on-Rails platform. A different webserver, such as Apache,[12] can be used in the future if more users need to access the webpage

- Windows Server 2008: This hardware was selected due to chemistry department's preference on Windows servers. However, other components of OSIRIS are platform independent, and can be installed on a Linux server without a problem.

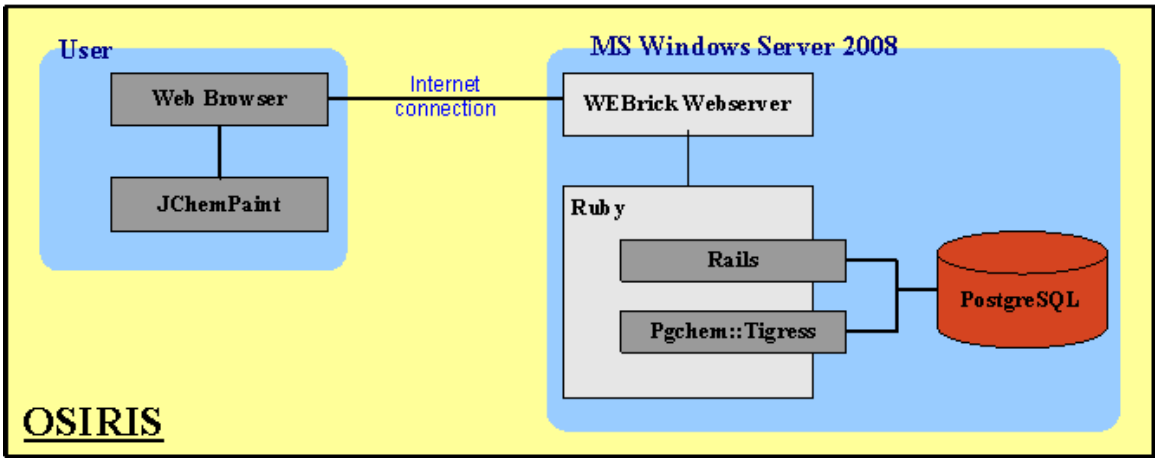



**Figure 1. Detailed schematic of OSIRIS requirements**

## *3.2 Logical Design*

OSIRIS has 3 levels of logical design (Figure 2). Level 1 is the user home page. Here, all users can access the "Shared Documents" folder, which distributes general literature for institute personnel such as manuals, procedures and articles from the literature, both scientific and non-scientific. Users also view projects to which they have access at this level. Assigning project viewing privileges is ordinarily managed by the system administrator through the web interface.

Level 2 of OSIRIS includes the project-specific files, all of which project members can share, access, upload and download. Again, the file privileges (Add Only, Manage Own, and Manage All) can be delegated from the web interface by the system administrator.

Level 3 focuses specifically on chemical compound management in a given project. Within this level, users can register new compounds and enter experimental results. Once a compound is registered, other users with proper privileges can add experimental results.

In EIDD, in order to preserve data quality and integrity the system is set up so that regular users can only add compounds, data, files, but only system administrator can edit/delete compounds, data, and files. Scientists can also use this level to create and submit ticketed requests for further compound studies such as BCDMPK (Bioanalytical Chemistry, Drug Metabolism and Pharmacokinetics) and solubility tests.

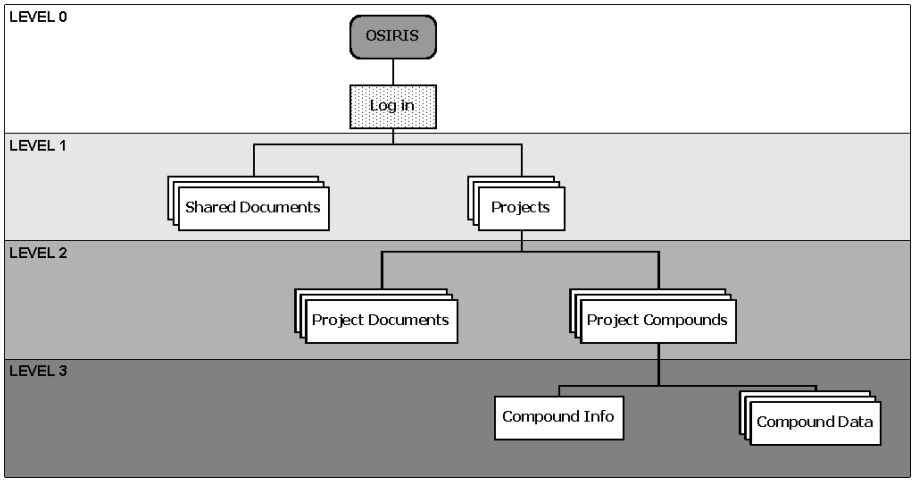



**Figure 2. Logical design of OSIRIS**

The logical design of OSIRIS was the first important step because it is the foundation of the whole program. After gathering requirements from stakeholders, creating the logical design of OSIRIS took a month of back-and-forth design changes between programmers and scientists. One of the biggest changes made was the addition of the "Shared Documents" to Level 1. At first, it was assumed that all documents would be put under their corresponding projects, but later it was obvious that there might be some documents that all users should have access to.

In the summer of 2010, a new module for ticket management for experiment requests had been added. This feature wasn't requested during the requirements gathering in spring 2009; however, it had been a year of users getting used to the program and scientists started to have ideas about how to make the system more useful for them.

## *3.3 Database Design*

A PostgreSQL database has been configured to allow maximum flexibility for adding new modules and features without compromising the rest of the database. A simplified Entity-Relationship (ER) diagram of the database design is shown in Figure 3. The main focus is the "molecules" table where each new compound is saved. Each user has unique project privileges, each project has multiple files and compounds, each compound can have multiple batches, and each batch can have multiple tests and files. This design lets the system administrator to create new projects and allows creating maximum flexibility on what each user can do in the system.

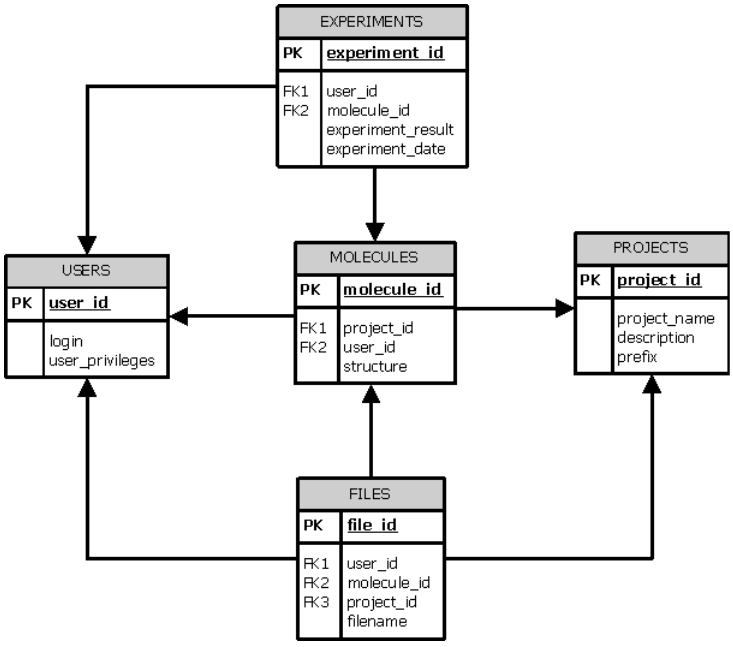



**Figure 3. A simplified representation of the database design.**

## *3.4 Data Types*

33 database tables and 152 database fields have been created. The most important tables and fields have been represented in Table 2. Automated index numbering has been enforced to provide consistent registration numbering assignment. Most of scientific data has been represented numerically so that it is easier to do custom searches such as using Boolean operators.

| Table Name | Field Name | Data Type |
|---|---|---|
| **Experiments** | user_id | integer |
| | molecule_id | integer |
| | experiment_result | integer |
| | experiment_date | timestamp |
| **Users** | user_id | integer |
| | login_name | string |
| | user_privileges | integer |
| **Molecules** | molecule_id | integer |
| | project_id | integer |
| | user_id | integer |
| | structure | string |
| **Projects** | project_id | integer |
| | project_name | string |
| | description | string |
| | prefix | string |
| **Files** | file_id | integer |
| | user_id | integer |
| | molecule_id | integer |
| | project_id | integer |
| | filename | string |

**Table 2. Data types for 5 main database tables**

## *3.5 Features*

### 3.5.1 User Privileges

OSIRIS has been designed to allow multiple user types to access particular projects, add files or register compounds. This can be controlled by the system administrator from OSIRIS's web-based control panel. The home page is customizable for every user based on their access privileges (Figure 4). Current user types include:

- User: add new compounds, add new files
- Project Leader: add/edit/delete new compounds, add/delete files
- Ticket Manager: approve/reject ticket requests
- System Administrator: create new projects, create new users, edit user privileges

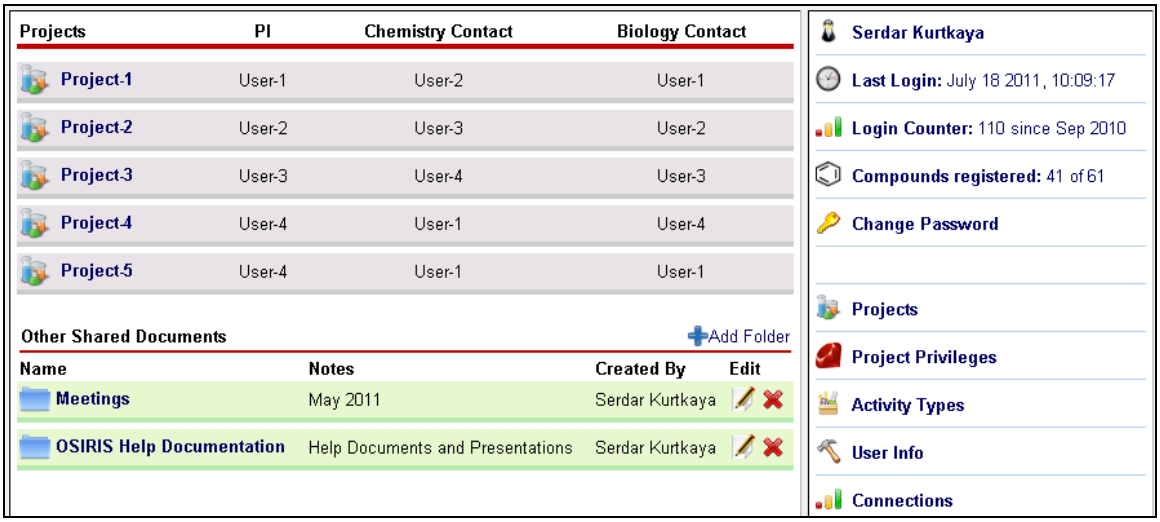



**Figure 4. Screenshot of the home page**

### 3.5.2 File Upload

Under each project, users can upload files based on custom designed file types. These include “presentations”, “chemistry” and “biology” (Figure 5). Users have the option to display all files for that project, or filter out only the desired category files such as “Biology”.

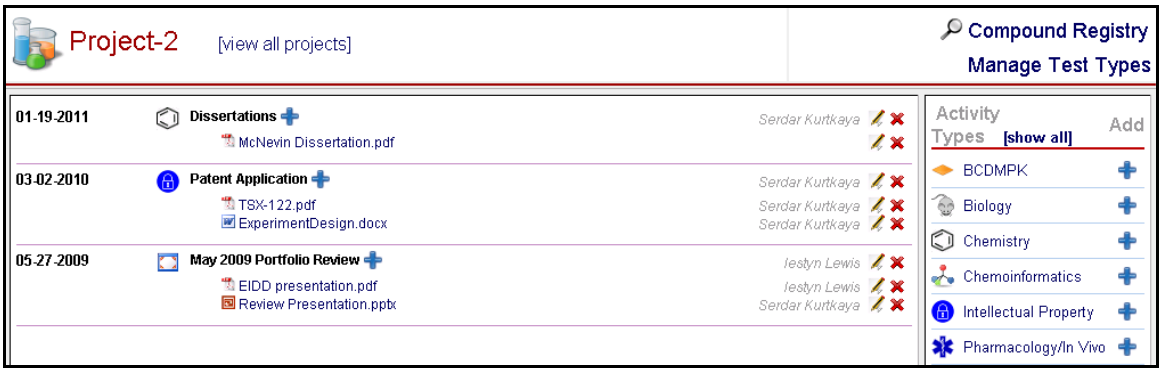



**Figure 5. Screenshot of the project view**

### 3.5.3 Compound Registration

Users can register new compounds to the system by means of interactive computer screen drawing of the structure by using the web-based chemical drawing tool (e.g. JChemPaint) or by importing a .mol file. The system assigns a unique 5-digit number to each compound structure preceded by a 4-digit project code. Thus, a particular chemical entity will have its own unique 5-digit number and a shared project prefix.

### 3.5.4 Data Management

OSIRIS has been designed to attach data to individual compound structures. Each project has a set of data tabs such as drug metabolism, analytical chemistry and pharmacokinetics (Figure 6). Each of these tabs is also accompanied by custom data fields such as $EC_{50}$, solubility and file name. This functionality is very useful for handling multiple projects with very different data fields. It is readily customizable to handle the requirements of a new project with unique needs or the changing direction of an ongoing project within a highly active research center.

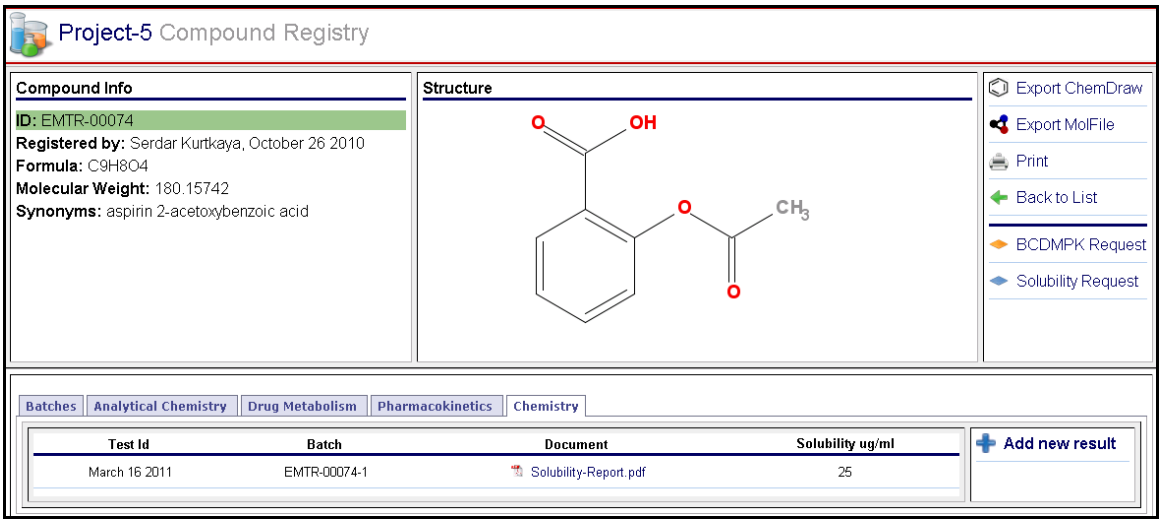



**Figure 6. Screenshot of an individual compound page**

### 3.5.5 Ticket Management

At the EIDD, scientists can have their compounds tested in the BCDMPK group, and in formulation and solubility laboratories. In order to reduce paperwork and prevent errors, a ticket management module has been incorporated into OSIRIS.

Users have the option to make a request from any service provider from the individual compound OSIRIS webpage describing the compound and its attributes. The request page automatically pulls the compound registration ID and compound molecular characteristics (structure, molecular weight, molecular formula). The requester just needs to fill out the rest of the fields such as, the nature of the request (pharmacokinetic analysis, compound stability analysis, etc.), and how much of a compound is being sent to the specialty lab.

This workflow also includes approval steps from the project leader and the requested lab supervisor. All requests are color coded to indicate where they are in the approval process, such as, approved, rejected and waiting (Figure 7).

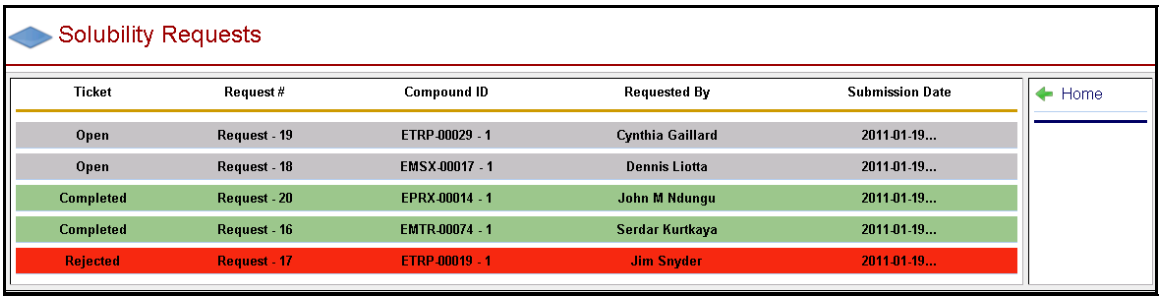



**Figure 7. Screenshot of solubility request ticket system**

### 3.5.6 Search Function

The system includes a search function based on a combination of substructure and text/data queries. Users can search for registration IDs, experimental result ranges, substructure of a compound, or any combination of all. The search function will only search within the project that the user has selected.

### 3.5.7 Data Export

All the structure and data within each project can be exported as an Excel or Structure Data (SD) file. Users also have the option to export individual compounds as .mol, .cdx or .sd file formats. The algorithm for data export is shown in Figure 8.

```
<%@mols.each do |mol|%>
    <%=mol.structure%> <%="\r\n"%>
     ><RegNumber> <%="\r\n"%>
     <%=mol.reg_number.to_s.strip%> <%="\r\n"%>
     ><Comments> <%="\r\n"%>

      <% mol.batchinfo.each do |b|%>
           <%= b.name.to_s.strip %> <%="\r\n"%>
      <% end %>

      ><Synonyms> <%="\r\n"%>

      <% mol.molecule_aliases.each do |a|%>
           <%= a.name.to_s.strip %> <%="\r\n"%>
```

```
    <% end %>
    <%="\r\n"%>

     <% @result_types.each do |rt| %>
          ><<%= rt.name.rstrip %>
          > <%="\r\n"%>
          <% results = mol.viewable_results[rt.id] %> <% if results %>
               <% results.each do |result| %>
                    <%= result.get_val.to_s.strip if result %> <%="\r\n"%>
               <% end %>
               <%="\r\n"%>
          <% end %>
          <%="\r\n"%>
     <% end %>
     $$$$ <%="\r\n"%>
<% end %>
```

**Figure 8. Algorithm for data export file**

### 3.5.8 User Statistics

System administrators have access to a toolbox that collects basic user statistics including measurement of the number of compounds registered, the total number of experimental requests made by each user and overall collection by all users. This feature is useful for figuring out who was online and what happened if there was a system malfunction.

## *3.6 Limitations*

OSIRIS had been created by only couple of programmers with limited sources. This resulted in number of limitations; such as:

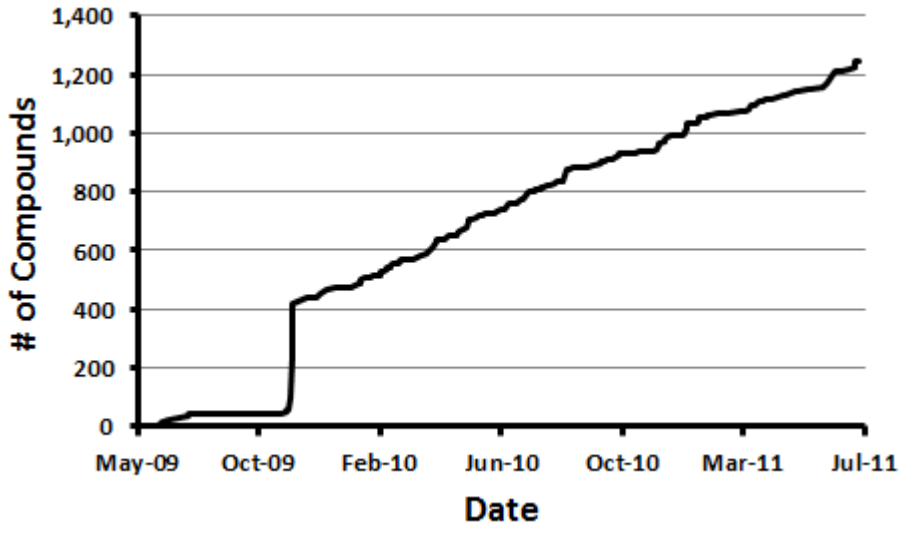
- EIDD does not plan on adding additional personnel to support the program even if there is interest
- Documentation of how to use the program is limited
- Ruby-on-Rails is not one of the most used programming languages; therefore, system administrator may have to learn a new programming language
- Ruby-on-Rails is slower on Windows servers than on Unix servers
- WEBrick webserver is good for small number of users (under 1,000). If more users need to connect at the same time, then more powerful webserver such as Apache webserver should be used.

# 4 Chapter 4: Results

Since its implementation in 2009, 1245 compounds have been registered on the system, and 608 project-associated files have been uploaded to occupy 2 GB of storage space.

## *4.1 Registered Compounds*

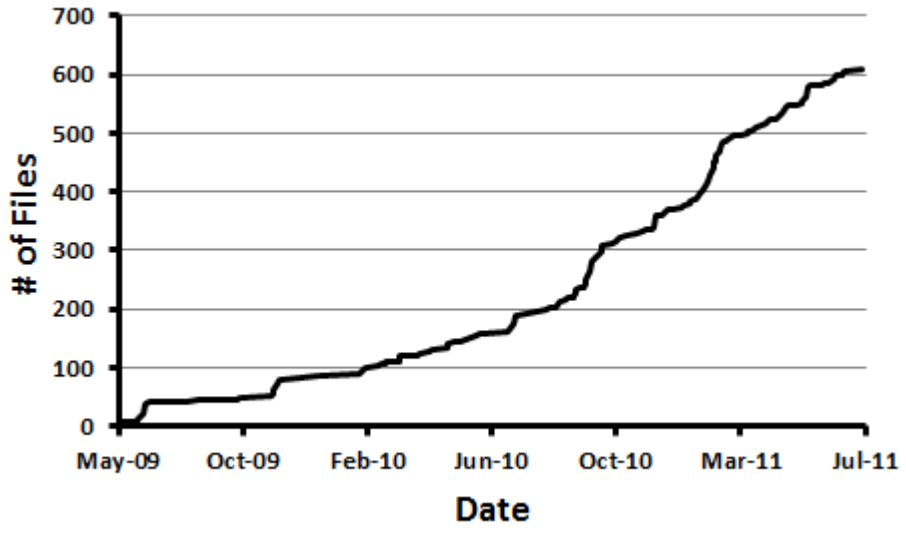
The number of registered compounds began slowly as users became familiar with the OSIRIS system. A total of 97 compounds were registered in the first 5 months followed by 321 compounds on three days in early November 2009 resulting from entry of data for two unregistered projects. This explains the otherwise unnatural jump in Graph 1. Since that time, the total number of registered compounds has increased linearly by approximately 42 new compounds per month.




**Graph 1. Total number of compounds registered over time**

## *4.2 Uploaded Files*

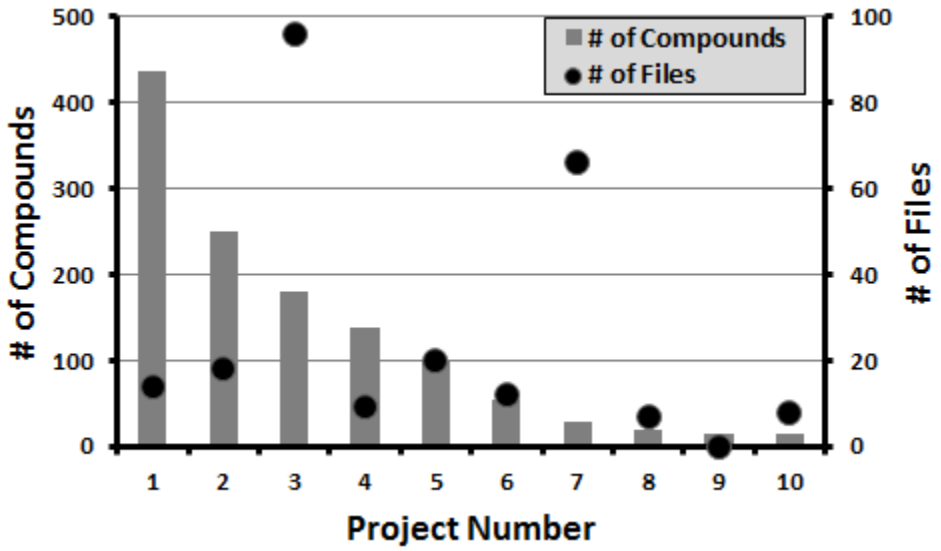
The uploading of project files was considerably faster than compound registration during the initial roll out stage of OSIRIS. The total number of uploaded files has followed an exponential curve with 12, 24 and 37 files/month in 2009, 2010 and 2011, respectively (Graph 2). The total size of the files has reached 2 GB, which is well under the storage capacity of the OSIRIS server.




**Graph 2. Total number of files uploaded over time**

## *4.3 Compounds vs. Files*

Surprisingly, the number of files uploaded in a project does not show a significant correlation with number of compounds registered (Graph 3). For example within 25 months following installation of OSIRIS, Project-1 has 437 registered compounds, however, only 14 uploaded files. By contrast, Project-7 reveals only 28 registered compounds and 66 uploaded files. One explanation for the ratio of registered compounds to project files has to do with the maturity of the project. Projects with a high number of uploaded files involve an increasing number of collaborators, causing the number of files to rise faster than the number of compounds registered.




**Graph 3. Project specific compounds and files**

# 5 Chapter 5: Conclusions, Implications, and Recommendations

## *5.1 Summary*

Tracking and linking relevant scientific data has historically been a challenge in drug discovery research. The Emory Institute for Drug Discovery has developed a system using open-source components to collect and use data generated in the Institute.

## *5.2 Conclusion*

OSIRIS has been developed since 2009. It is important to note that the web-based and platform-independent nature of OSIRIS has resulted in successful user adoption. The number of compounds registered and files uploaded has increased consistently over a two-year period. The ability to share raw as well as correlated data and associated files has been a great asset in a multi-project institute. The EIDD members have been using the system to efficiently share data in an organizational structure where the various disciplines of drug discovery are unique and independent entities. The open-source nature of the program also resulted in very flexible customization of the capabilities of the system.

## *5.3 Implications for Practice*

This program can be used in variety of public health settings, especially in the Environmental Health area. Since OSIRIS already has an integrated chemical structure viewer, synthetic or natural compounds, toxins, and toxicants can be entered to the system.

It is the intention of the EIDD to provide the source code of the system to the academic and non-profit community free of charge. This should be very helpful in Public Health area where monetary sources can be scarce. Any non-profit public health institute can download the source code and modify it to work the way they require.

## *5.4 Recommendations*

The lack of documentation has been a major drawback. In order to ensure higher acceptance rate, program documentation should be completed and provide ongoing system knowledge should the programmers leave the organization. A detailed explanation of how different open-source tools are installed and connected should be the first action.

Currently, there is no website to promote the software. A dedicated webpage with screenshots and demo versions should be implemented.

# 6 References


1. Emory Institute for Drug Discovery. http://eidd.emory.edu.

2. Kerwin, S. M., ChemBioOffice Ultra 2010 Suite. *J Am Chem Soc* **2010,** *132* (7), 2466-2467.

3. Open-LIMS. http://www.open-lims.org.

4. OpenELIS. http://www.openelis.com/.

5. WEBrick. http://www.webrick.org/.

6. PostgreSQL. http://www.postgresql.org.

7. Ruby on Rails. http://www.rubyonrails.org.

8. pgchem::tigress. http://pgfoundry.org/projects/pgchem/.

9. Krause, S.; Willighagen, E.; Steinbeck, C., JChemPaint - Using the collaborative forces of the Internet to develop a free editor for 2D chemical structures. *Molecules* **2000,** *5* (1), 93-98.

10. PHP Hypertext Preprocessor. http://www.php.net/.

11. Perl. http://www.perl.org/.

12. Apache HTTP Server. http://httpd.apache.org/.