Lessons from the Development of SSEUS: a System for Entry and Management of Peer-reviewed Data

Susan Jeffery
Jeffery Systems
susie@unicon.sf.net

Clinton Jeffery
University of Idaho
jeffery@uidaho.edu

Phillip Thomas
National Library of Medicine
ptho@firefly.nlm.nih.gov

Abstract

Specialized information Services Edit Update System (SSEUS) is a peer-reviewed data entry management system that is used to maintain several databases at the National Library of Medicine at the National Institutes of Health. SSEUS pairs a MySQL database server with an elaborate, open-source client written in an open-source very high level language. This paper presents lessons learned during the development, deployment, and maintenance of SSEUS over a ten year period. Compared with a proprietary system that it replaced, the use of open source technologies has reduced costs substantially, while improving the flexibility and maintainability of the databases.

1. Introduction

In 1999, the National Library of Medicine Specialized Information Services Division began a migration of several of its databases from closed-source vendors to open source solutions. Ten years later, we are able to report on some of the impact of this switch.

The previous TOXNET system involved client pay-for-search over dial-up connections. Depending on the information requested, responses were obtained from either a set of networked DOS machines or from an IBM mainframe. Data going into the system were converted from disparate sources delivered on tape and then processed in MUMPS code passing to the networked PCs or was processed on the mainframe with code written in PL/1.

With the availability of the Internet and a Government directive to provide the data for free, a partially successful system replaced the networked PCs with a pair of Sun Solaris machines. The development group had only a rudimentary understanding of UNIX programming and security. The original RDES (Remote Data Entry System) written in Visual BASIC was redirected at this platform. Navigation was difficult. Results usually had to be emailed to those querying the system. With single records reaching up to 150 pages, the mail server often had to be restarted. Database synchronization and backup required the servers to be taken off-line one day a week.

Here are some of the problems which led us to changing our previous TOXNET system to the current one:

1. The old data update system (RDES, Remote Data Entry System) was written in Visual BASIC and could not handle sufficiently long strings. Individual string data entries in health information databases such as the Hazardous Substances Data Bank HSDB run routinely into the 1MB range.
2. RDES worked natively through a direct dial-up connection and only with third-party add-ons through the newly emerging Internet connections.
3. The backend MUMPS database code had few people able to read, debug, or improve the code. New Internet demands and expectations required an entire rewrite of the database code to whatever data entry solution was chosen.
4. Printouts of incoming TOXNET data for the Scientific Review Panel could not be easily modified. The print function worked only on high-speed line printers which printed single-side on fan-fold paper. The SRP sessions thus required some 5,000 sheets of paper for each participant. It was highly desirable to print double-sided, stapled packets.
5. The mainframe on which the MUMPS database ran was about to be retired.

These issues motivated the development of a replacement system under the acronym SSEUS, standing for Specialized Information Services Edit/Update System. Other than being an homage, it

IEEE computer society

has nothing to do with the popular children's author, whose name is spelled slightly different.

2. Requirements

SSEUS's basic requirements were to perform the numerous database reviewing and updating tasks that were previously the job of its VB-based predecessor, RDES. The SSEUS system is made up of a client, a database server, and numerous back-end special purpose programs that handle jobs such as data migration or batch printing. The reader can envision a classic business database application for which a RAD (Rapid Application Development) language was suitable. Then imagine that application growing and mutating well beyond the domain for which such languages are suited, with extensive and ongoing additions and changes to the requirements.

Consequently, the first requirement for SSEUS was an applications language with very high level capabilities not just in its user interface building and database connectivity, but its ability to implement new algorithms on large datasets, particularly large text strings. With the rapid software development capability provided by a very high level language, new features get to the users faster. These features reduce data entry time which is critical to lower costs and provide the online system with the most current data.

The output of SSEUS is a copy of the edited records, which are migrated from the internal review database to public status. The data is exported from MySQL in SGML format to a public retrieval system written in C using the PISCES database libraries. The information retrieval task is not in the scope of this paper. Export of data in SGML format is one of the simpler requirements for the SSEUS system.

The primary tasks supported by SSEUS revolve around the data input process: it is used to facilitate the complex multi-stage peer review of updates and additions to chemical information obtained from a multitude of published scientific sources. The data comes in many languages. A given chemical has dozens of names due to differences in language, scientific discipline, and use. A major requirement for SSEUS was support for complex data input checking rules that vary on a per-field basis.

The chemical information must be arranged in sections for display and update, for example: Physical Properties, Safety and Handling, Pharmacology, References. Each section contains the fields that contain the data. For example, the section "Chemical Properties" contains the fields: "Odor", "Taste", "Boiling Point", and "Molecular Weight". Each of these fields can contain a bibliographical reference and comments and recommendations. There can be up to 150 of these fields for each chemical and some of these fields are larger than 360 KB.

Users are members of one of several SSEUS groups, each with different roles and update rights (Figure 1). The users have different security requirements based on their function. For example, the "File Builder" user group generally consists of the government contractors who input all the chemical information. The "SRP" (Scientific Review Panel) user group cannot modify the chemical data, but can add comments. The "DBA" (Data Base Administrator) can commit the chemical data to go online after it has been reviewed by the SRP.

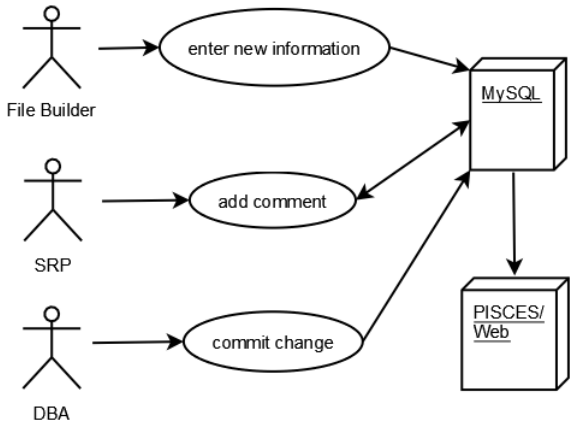

Figure 1: SSEUS's three classes of users

TOXNET contains different categories or databases for the chemical information. SSEUS currently supports HSDB (Hazardous Substances Data Bank), CCRIS (Chemical Carcinogenesis Research), DART (Developmental and Reproductive Toxicology), and LactMed (Drugs and Lactation). There are more than 5,700 chemicals in HSDB, 9,000 chemicals in CCRIS, 32,000 in DART and 700 in LactMed.

3. Design

The software is organized into program modules (files) based on functionality. Most of these files are dialogs that branch off the main screen by selecting functions from the menu on the main screen. Figure 2 is a screen shot of the dialog of the SSEUS main program.

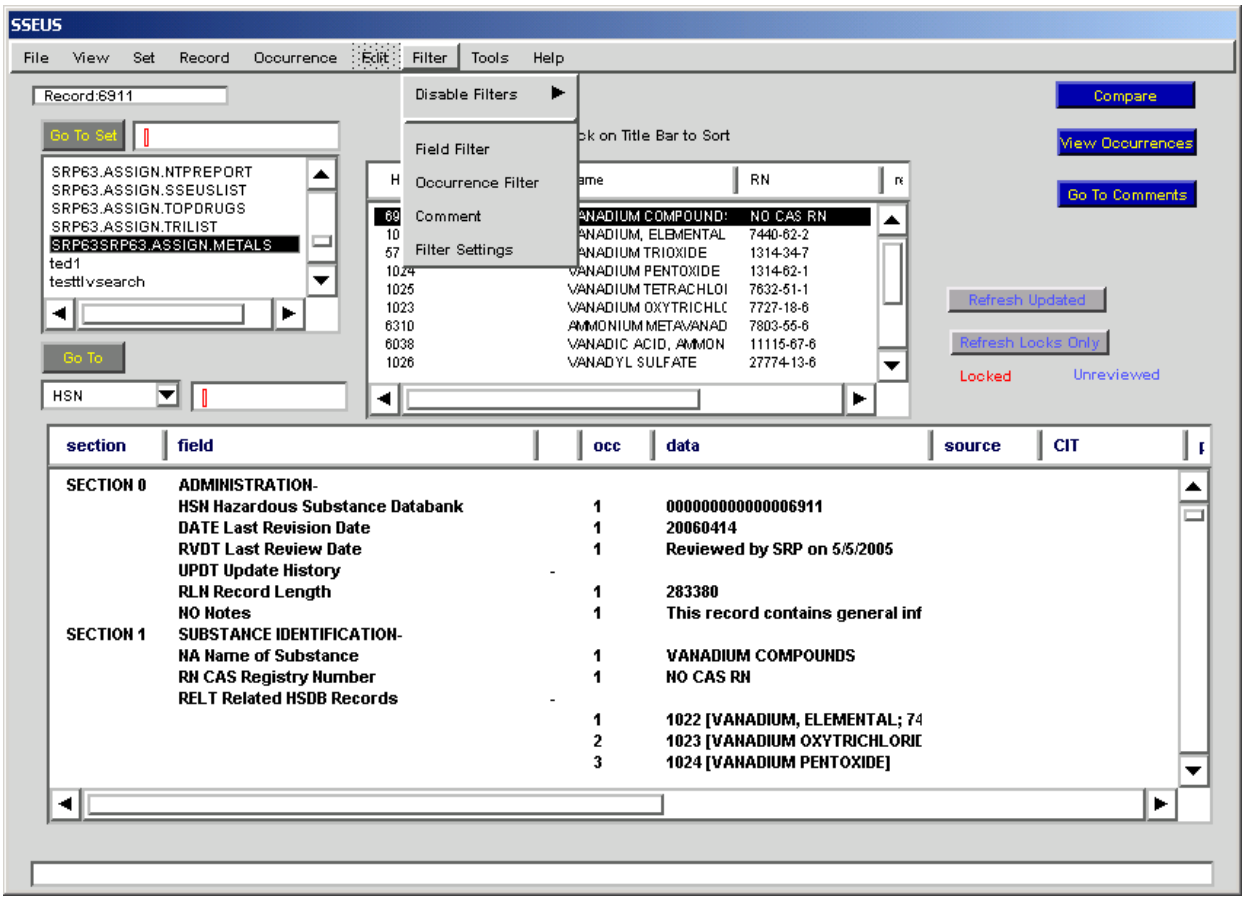

Figure 2: The SSEUS client

3.1 Client Module Organization

From the main program, you can select a chemical and a field to edit or view. The drop down Filter Menu item shows different options for filtering the data for display. Almost all of the options from the file menu on the main screen have a dialog associated with them. Some example program modules (dialogs) are field_filter, occurrence_filter, search, create_set, and copy_occurrence. This modularity provides for easy maintenance and addition of enhancements. There are over 40 different calls to MySQL within SSEUS. The majority of the MySQL calls are in one file for ease in debugging and modification.

Figure 3 shows a more detailed view of SSEUS's use cases, which are reflected in the primary dialog classes of the design.

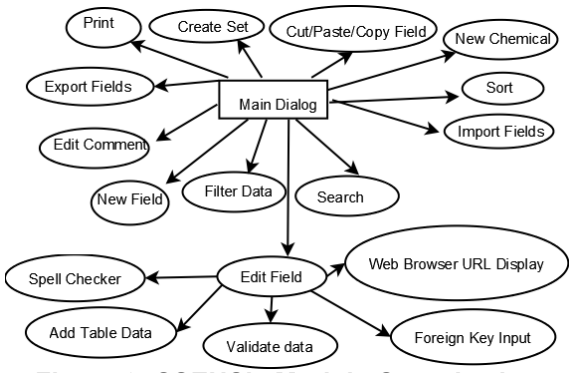

Figure 3: SSEUS's Module Organization

MySQL does not support record locking to prevent users from updating a record at the same time. Since 20 users could be updating a table at the same time, SSEUS must provide its own locking. SSEUS locks the entire chemical or a specific field depending on the transaction. It inserts the chemical or field in the lock table during the update and removes the entry from the table after the

transaction is completed. SSEUS checks the lock table before each transaction to make sure the chemical or field is not in use.

Because the data resides in the open source MySQL database, it is easy to maintain. Data is easily imported from outside sources to MySQL by batch Unicon programs. Unicon programs are also used to transfer the data from MySQL to SGML for the online system. Unicon programs are used for printing reports in different formats.

3.2 Relational Database Design

The data is organized in a relational database containing roughly 10 tables. The frame table contains the order of display of the fields and sections. The following is a snippet of the data from section 3 in the frame table:

Name	Field	Section
Chemical and Physical Prop.	CPP	3
Color/Form	COFO	3
Odor	ODOR	3
Taste	TAST	3
Boling point	BP	3
Melting Point	MP	3
Molecular Weight	MW	3

The Chemical table contains the chemical number and record number. The following is a snippet of data from the chemical table:

Rec	Name
0012	STRONTIUM SULFIDE
0031	ISOLAN
0033	MERCURIC CHLORIDE
0034	IODINE, ELEMENTAL
0035	BENZENE
0036	CAFFEINE
0037	CAMPHOR
0038	CETYLPYRIDUNIUM
0039	EPICHLOROHYDRIN
0040	ACETIC ACID
0041	ACETONE

The Field table contains the chemical information and references. The following is a snippet of data for chemical number 0040 (Acetic Acid), field mw (Molecular Weight), and Source number 027.

Rec	0040
Field	Mw
Data	60.05
Source	27
User	Joe
Indate	2004-02-17 15:24:49
Keyn	3944

The Source table contains the source number and the reference information. The users enter the Source number for each field and this number is linked to this table to display the source information. The following is a snippet of data for Source number 27:

No	Source
0000027	LaDu, B.N., H.G. Mandel, and E.L. Way. Fundamentals of Drug Metabolism and Disposition. Baltimore: Williams and Wilkins, 1971.

The Comment table contains the comments associated with each field. There can be more than one comment per field and it can be entered by different members. The following is a snippet of a comment for field mw in chemical record number 0040.

Rec	0040
Field	Mw
Keyn	3944
Indate	2003-04-05 12:38:38
User	Joe
Comment	Comment 1: Info not in source

The Set Table tracks collections of chemicals that are being worked on by individual file builders or scientific review panel members. These sets are submitted to the DBA for approval when they are completed so they can be moved to the online system. The following snippet is a set of chemicals gathered logically together for review under the set name “drugs". The table contains the chemical rec numbers.

Setname	Rec
drugs	0033
drugs	0034
drugs	0036
drugs	0037
drugs	0038
drugs	0040
drugs	0046

The Lock Table contains the user and chemical fields so only one user can update a field at a time.

Keyn	User	Rec	Field
3944	Joe	0040	mw
21	Mary	0035	bp

Figure 4 shows the primary relationships between tables.

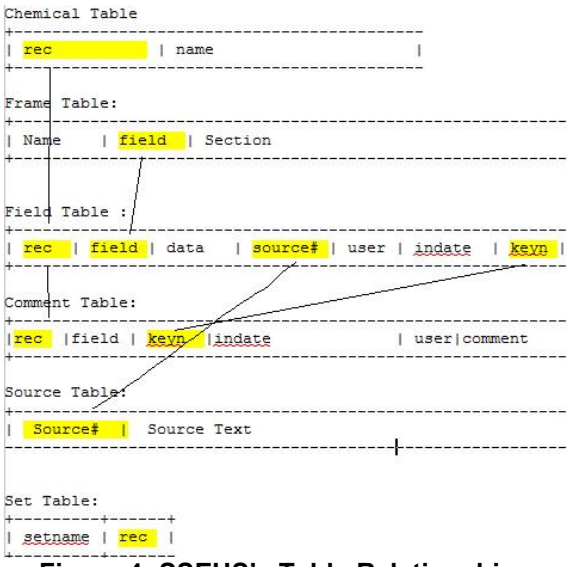

Figure 4: SSEUS's Table Relationships

4. Implementation

SSEUS is written in Unicon which is a very high level language [1]. Unicon supports features such as associative tables, sets, sorting, searching and string processing as built-ins. It minimizes programming time which is one of the most expensive resources in any application. SSEUS's implementation evolved over several years.

The original plan was to make SSEUS as generic as possible so that the same code could support the different databases (HSDB, DART, CCRIS and so on). Unfortunately, this was not possible because each database had very specific requirements. Therefore, the common features are supported by the same programs, but separate modules (programs) support the different features of the databases.

4.1 User Interface

The SSEUS graphical interface was written using IVIB (Improved Visual Interface Builder; see Figure 5). IVIB is a program for designing GUIs using a component drag-and-drop interface. Dialogs with components such as buttons, menus, text boxes can be created quickly using IVIB. Designs are output as Unicon source code, which can be subsequently edited using both the IVIB tool and regular programming tools. Close to 30% of the SSEUS code is automatically generated by IVIB. The other 70% constitutes the reason for using Unicon instead of, say, Java or Visual Basic.

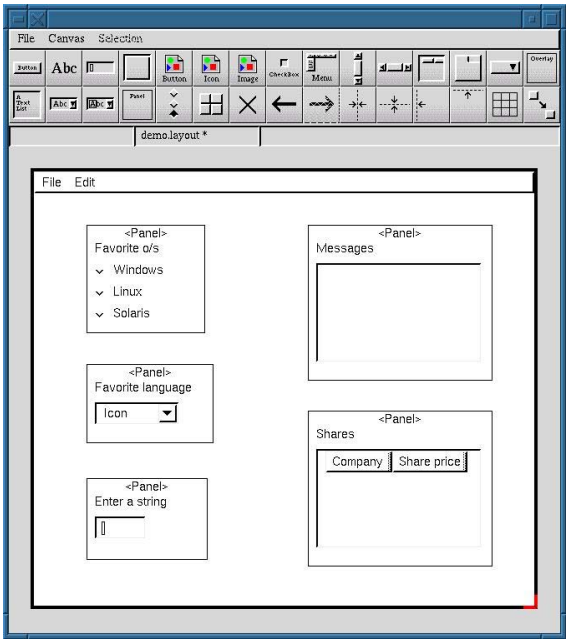

Figure 5: A SSEUS dialog edited with Ivib

4.2 Database Connectivity

The Unicon language was extended with ODBC capabilities specifically for this project [2]. The chemical information is stored in a MySQL Database residing on a Linux server using MySQL Server version 5.0.77. The database connection to the server is made using MyODBC version 3.51. On Windows the ODBC configuration is automatic; on Linux, the ODBC configuration must be added to the .odbc.ini file.

SSEUS connects to the server through a tunnel using PuTTY, an Open Source SSH client. The SSH tunnel provides sufficient security. The SSEUS software accepts the MySQL user and password and opens the database.

4.3 Domain Functionality

Because the Unicon and IVIB code is open source, it was possible to customize SSEUS extensively to support specific functions for chemical applications. SSEUS supports many special functions. It has a Spell Checker with a Dictionary that contains medical terms and chemical symbols. Latin II and other non ASCII characters may be entered to support foreign authors and

chemical units. Different colors and highlighting of text are used to enforce correct data entry. Validation checks and calculations on numerical data are done before the data can be saved. Control sequences and buttons are used to enter predefined phrases to reduce key strokes, ensure consistency and correctness.

SSEUS's predecessor, the RDES system, required knowledge of a particular chemical or set of chemicals. For example, the first thing users input was a chemical name. Since SSEUS programmers did not have a chemical background, SSEUS was developed to "Select and Go" instead of entering predefined data. For example, In RDES, users first type in the particular database (such as HSDB, CCRIS or DART) and then type in the chemical name (e.g. Benzene, Caffeine, Chlorine etc.) With SSEUS, users select a database from a scrollable list of databases. After the database is selected, users select a chemical from a list of chemicals.

SSEUS supports a variety of Search options. Users can search for words or phrases using Boolean "and", "or", "not" and searches can be restricted to specific fields or sources. The Search dialog is written using MySQL queries combined with Unicon code because the required searches are too complex to be performed by MySQL queries alone.

5. Deployment

The SSEUS client program runs on Microsoft Windows XP, Vista, and any reasonable X11 Workstation such as Linux. The SSEUS downloadable binary installer includes the SSEUS program. For Windows, it also includes the installer for the MyODBC driver and the SSH tunnel.

After connection, SSEUS checks the client version of SSEUS against the current version of SSEUS on the server. If the server's version of the client code is newer, it is downloaded to the client. The client code is stored on the server in an actual MySQL table. The virtual machine executable and the byte code for the client are maintained in separate files in order to reduce the size of the typical patch download. This provides easy patching and enforces version control. Because SSEUS validates input and enforces rules that constantly change, it is essential that all users be using the same version. The patch technique was generally inspired by the practice of MMORPG game software, although such practice has become common in other software categories since then.

Because each chemical contains so much information, SSEUS supports different views, data filtering and sort options. SSEUS includes a dialog to create sets of chemicals that are often assigned to a user or group of users. After the sets are approved, SSEUS marks the records approved and moves the chemicals to the production table so they can be written in SGML for the online system.

There are two tables for HSDB, CCRIS, and DART with the same table design: the maintenance table and the production table. The maintenance table is updated by users. The production table is never updated by users. Once chemicals are approved in the maintenance table, the chemical is "moved" to the production table. A Unicon program copies the chemical from the maintenance table to the production table. A Unicon program translates the MySQL data from the production table to a SGML file for the online system. Thus, the data in the production table reflects the data in the online system. The chemical information is exported to SGML for display on "TOXNET", the Toxicology Data Network for NLM.

6. Software Process

The software process used to develop SSEUS can be characterized as agile, informal, and demand-driven. An initial working version of SSEUS was derived by reducing the specifications of its predecessor to its most essential features. From then on the process was iterative. Each development cycle started with a user request for a new feature. The specifications were discussed by phone and e-mail. Sometimes, management intervened, usually to veto a requested feature that was vestigial in the legacy predecessor system and non-essential. New features were implemented rapidly and released using the SSEUS patcher (see Section 5). Each feature was polished through multiple iterations of feedback and revision until the users were satisfied.

The majority of the SSEUS system was built by a single programmer with no domain expertise and no formal training in software engineering. It is reasonable to ask why this process succeeded. The size, complexity, and technical domain of the project would seem to necessitate a much larger team with a commensurate budget, and formal documentation of each software development phase.

One reason for the process' success was the competence of the individuals involved. Constant, close communication between the developer and the users and domain experts overcame domain difficulties. This close communication was possible because the number of persons was kept small, which was possible because of the very high level language used.

Another reason for the process' success was that SSEUS's predecessor, RDES, constituted a detailed prototype that both enumerated many requirements and illustrated many pitfalls to avoid. Frederick

Brooks' aphorism *Plan to throw one away* [3] was satisfied in large part by RDES.

Much of the original SSEUS design was presented as a series of screen shots using a standard Java interface builder. Although the domain experts were not familiar with the very high level implementation language, they were able to develop these screen shots to represent the user requirements in a visual form. For example, Scientific Review Panel members needed to see the chemical data and all comments associated with that data on one screen.

7. Metrics

SSEUS has reduced expenditures significantly (in the ballpark of one or more orders of magnitude). Precise measurements there are impossible because tasks performed by SSEUS and its predecessor are non-identical. The main goal was not cost reduction, but increased control over the project and ability to make changes quickly that previously were nearly impossible and very expensive.

7.1 Space

There are two measures of space which matter for SSEUS. On the server, the disk space needed by the databases is a function of MySQL and not under our control. The SSEUS client is a medium-sized program that handles large amounts of data. Low-end systems have been known to have problems due to lack of sufficient main memory. Part of the trade-off of a very high level language is that its flexible memory representations are not as compact as an optimally coded solution written in a lower level systems programming language such as C or C++. The difference between running well on a 256MB machine or a 512MB machine or a 1GB machine has largely been erased by large desktop and laptop main memory sizes.

7.2 Time

SSEUS as a MySQL client has had a few performance issues. MySQL performance can be extremely slow for various queries necessary to process the chemical data. Some of the slow MySQL processing time was improved by adding tables. For example, the chemical table was created to only include the chemical name, number and registry number. This information is also in the field table. Since the field table contains approximately 1,417,000 records and the chemical table contains only approximately 5,000 records, extracting the chemical name, number and registry number from the field table was much faster. However, SSEUS must ensure that these fields are in sync in both tables. Performance for MySQL single select and fetch operations was poor. SSEUS compensated for this by fetching more information on a select statement. The information was stored in memory until it was needed. There is still slow response time for certain SSEUS functions such as the exporting of multiple fields from multiple chemicals.

Another performance issue worth reporting was from the simultaneous batch addition of Emergency Medical Treatment data while SSEUS users were working online. This was solved mainly by rewriting the batch loader for the EMT data.

There is a related general, non-SSEUS-specific performance issue that is unsolved: how to spread MySQL across several processors using the current database type. Users, batch uploads, and sub processes are all using the same processor even though we have 4 processors available, giving a typical server load of 99% on one processor and only a few percent on the others.

8. Related Work

One obvious area of related work is that of peer-review systems used by scientific journals and conferences. The ESPERE Project is an example of such a paper refereeing system [4]. A basic difference between SSEUS and such systems is the corrective, rather than judicial nature of the peer review. The input sources are generally already accepted published materials, which contain information that may add to or correct existing information in the national databases. The Scientific Review Panel or NLM staff scientists are not typically playing an "accept" vs. "reject" role, but rather, one of correction and conflict resolution. SSEUS's database manages new data through a pipeline on the order of 15 stages on its way into the public databases.

Closer to SSEUS are the systems used by other databases with extensive peer-review of their content, such as Clinical Pharmacology [5], and the Natural Medicines Comprehensive Database [6]. Most of these databases will have published descriptions of their peer review process, without documentation of the system by which data is managed for comparison with SSEUS. A happy exception is the Molecule Pages database SGMP [7]. This database uses a combination of peer-reviewed, expert author data along with automated data collected from public bioinformatic data sources and sequence results. SGMP is no doubt representative of many related works, in that its

client consists of web forms that connect to enterprise Java servlets to fetch data from an underlying database (in SGMP's case, Oracle). This approach was considered and deemed inadequate for SSEUS because of the extensive custom input checking requirements.

8. Conclusions and Future Work

SSEUS replaced an expensive proprietary system maintained by a team of government contractors with an open system written and maintained largely by a single person, interacting with appropriate NLM system administrators and staff. Because the source code of the system -- not just the application source code but the underlying programming language source code -- is delivered to the government, the government is in more control of its costs and features.

The success of SSEUS is evidence in favor of requirements-driven, agile development. A complex system was assembled without a large team or a heavyweight software engineering process.

The user requirements and validation rules for SSEUS change continually. Therefore, the most important feature for SSEUS was probably the patcher program. Software changes need to be done quickly on a daily basis. Therefore, good software organization and design were instrumental to quick turnaround time.

SSEUS has been installed on Macintosh Computers with a lot of tweaking. The Macintosh installation and support will be improved and stream lined in the near future. It helps dramatically that Apple now ships the X Window System standard on OSX.

9. Acknowledgments

This research was supported in part by the Specialized Information Services division of the National Library of Medicine, and by the NLM Research Participation Program. This program is administered by the Oak Ridge Institute for Science and Education for the National Library of Medicine.

10. References

[1] Clinton Jeffery, Shamim Mohamed, Ramon Pereda and Robert Parlett, *Programming with Unicon*, book and software downloads from from unicon.org, 2007.

[2] Federico Balbi and Clinton Jeffery. *An ODBC Interface for the Unicon Programming Language*. Unicon Technical Report 1b, Unicon.sf.net, 2002.

[3] Frederick Brooks, *The Mythical Man-Month: Essays on Software Engineering (2nd Edition). Addison-Wesley, 1995.*

[4] ESPERE Project. www.espere.org.

[5] Clinical Pharmacology. www.clinicalpharmacology.com.

[6] Natural Medicines Comprehensive Database, www.naturaldatabase.com.

[7] Brian Saunders, Stephen Lyon, Matthew Day, Brenda Riley, Emily Chenette, and Shankar Subramaniam. *The Molecule Pages database.* Nucleic Acids Research. Oxford Journals. http://nar.oxfordjournals.org/cgi/content/full/gkm907v1.