IT DEVELOPMENT AND MANAGEMENT OF A LIVE E-RESEARCH SYSTEM

Experiences with the Australian Schizophrenia Research Bank

Paul, D.[1,2], Henskens, F. A.[1,2], Loughland C. M.[1,2], Bridge, J.[1,2], McCabe K.[1,2], Carr, V. J.[1,3], Catts, S. V.[1,4], Jablensky, A.[1,5], Michie, P. T.[1,2], Mowry, B. J.[1,4], Pantelis, C.[1,6], Schall, U.[1,2] and Scott, R. J.[1,2]

[1]*Schizophrenia Research Institute, Sydney, NSW, Australia*
[2]*University of Newcastle, Callaghan, NSW, Australia*
[3]*University of New South Wales, Sydeny, NSW, Australia*
[4]*University of Queensland, Brisbane, QLD, Australia*
[5]*University of Western Australia, Perth, WA, Australia*
[6]*University of Melbourne, Melbourne, VIC, Australia*
Corresponding author: David.Paul@newcastle.edu.au

Keywords: eResearch, Schizophrenia, ASRB

Abstract: The Australian Schizophrenia Research Bank (ASRB) is a nationwide eResearch project that aims to facilitate scientific research into one of the most perplexing and challenging mental disorders facing researchers today. The system is accessed through a Web portal and, subject to ethics approvals, allows researchers access to subsets of the clinical, neuropsychological, and genetic data collected by the project. This paper describes the day-to-day experiences in the on-going development and management of the ASRB IT systems, including current practices, lessons learned, and areas where improvement is required.

1 INTRODUCTION

Schizophrenia is a mental disorder that affects approximately 0.3-0.7% of the population (Os & Kapur, 2009). Despite this low prevalence, the burden of the illness upon sufferers and their families, and society in general, is extremely high. The disease is characterised by cognitive, perceptual, and affective dysfunctions, often presented through hallucinations, delusions, and social withdrawal.

The cause of schizophrenia is unknown, though evidence does suggest that genetic, environmental, and social factors all contribute (Mueser & McGurk, 2004). There is no known cure, but the disease is typically treated using antipsychotic drugs and psychosocial treatments. Most patients can typically live independently outside of hospital, though they often require some degree of financial or daily living support.

The Australian Schizophrenia Research Bank (ASRB), previously described in (Henskens, et al., 2009), is a nationwide (to Australia) collaboration of scientists that aims to facilitate scientific research into schizophrenia by collecting, storing, and providing a large set of clinical, neuropsychological, genetic, and brain imaging data from both people with schizophrenia and healthy controls. The project began in 2006, and has currently collected clinical interviews, MRI brain scans, and/or blood samples from over 1500 volunteers.

The electronically storable components of the data collected for the ASRB, and meta-data describing physical samples, are entered into a central system through a Web portal. This allows the data to be accessed by any authorised Internet-connected researcher, without the requirement of high-powered client-side equipment.

This paper reports on our experiences with the ASRB IT systems, providing valuable insights for future developers of similar health-related research support. An overview of the ASRB system is given in Section 2. Section 3 outlines its major functionality. Section 4 describes the general process used in the system's maintenance. Sections 5 and 6 then discuss application development and database management respectively. Finally Section 7 concludes this work by reviewing the lessons that

have been learnt and discussing potential future improvements.

2 SYSTEM OVERVIEW

The ASRB system is accessed through a Web portal that securely allows data collection and access. The system manages all phases of data collection from first contact with a potential volunteer, through a comprehensive clinical interview, and on to the collection of genetic samples and brain images. Once the data is in the system, authorised researchers are granted access to the subset of data that they have approval (from both an Ethics Committee and the ASRB Access Committee) to view.

The server uses the Liferay enterprise open source portal (Sezov, 2009) and provides portlets for managing participant intake; clinical assessment; and the collection of brain images and genetic samples. Portlets that offer query support are also provided, allowing researchers to discover which parts of the data that they may wish to access.

Most of the data collected for the ASRB is entered directly through the server portal. A notable exception to this is the clinical interview, which is typically conducted 'in the field' with data entered into a laptop; the interview data is later securely uploaded to the central server. The reason for this is that the interviews often occur wherever it is most convenient for the participant, meaning that there is no guarantee that an Internet connection will be available during the collection of an interview.

To allow the clinical assessments to be collected without requiring a connection to the Internet, the clinical assessment portlets are deployed on an Apache Pluto (Hepper, et al., 2005) portlet container installed on the laptop. Pluto is used on these systems because it has sufficient features to support the clinical assessment portlets and has lower resource requirements than Liferay. This is an important consideration because the laptops have much lower specifications than the server system.

On both the server and the laptops, the ASRB data is stored in a PostgreSQL database (PostgreSQL Global Development Group, 2011), typically accessed through the Java Persistence API (Biswas & Ort, 2006). This allows general access to the data through the Web interface, but it is also possible to connect directly to the database to diagnose or repair any problems that occur.

Development of the ASRB follows a typical test-driven development process (Beck, 2003) and thus allows fast turnaround for requested enhancements or required bug fixes. This is facilitated via a Jenkins Continuous Integration server (Kawaguchi, 2011) that supports automatic test execution whenever code is committed to the ASRB Subversion (Collins-Sussman, Fitzpatrick, & Pilato, 2004) repository. The integrated tests include code style checking through Checkstyle (Burn, et al., 2010), unit testing with JUnit (Beck & Gamma, 2011), and test coverage using Cobertura (Doliner, 2006).

3 SYSTEM COMPONENTS

The functionality of the ASRB system is divided into a set of portlets that split the ASRB Web application into smaller components. Access to each portlet is restricted to users in particular roles. For example, users granted the Clinical Assessment Officer (CAO) role have access to the portlets that manage participant intake and clinical assessment, but do not have access to the blood collection portlets, which require the Blood Technician role. This section describes the main portlets used for the ASRB.

3.1 Participant Intake

The intake portlets are used to enter and manage the details of potential participants in the ASRB project. First contact with potential participants is via either an expression of interest form, or through a telephone call. In either case, the portlets provide a guide that can be followed by CAOs and other authorised users to enter the new volunteer into the ASRB system.

Not every volunteer will become an ASRB participant, but the intake portlets allow information to be tracked to help determine which volunteers should be included in the project. Once a volunteer has been accepted as an ASRB participant, the portlets generate a unique id for the participant and allow the tracking of the participant's progress through the clinical assessment, MRI scanning, and blood sampling processes of the project.

3.2 Clinical Assessment

The clinical assessment process of the ASRB project is an extensive interview with the participant that includes: the detailing of socio-demographic and clinical history; neurological, personality, and cognitive functioning evaluations; a psychosis screener; and the Diagnostic Interview for Psychoses

(DIP) (Castle, et al., 2006). The clinical assessment portlets allow the entire interview to be entered electronically while the CAO is interacting with the participant. This ensures that questions are asked in the correct order and allows customised navigation, so that questions irrelevant to the current participant are automatically skipped. Electronic collection also enables scores to be calculated and verification of data to occur as the interview progresses, which significantly reduces human error by immediately alerting the CAO of any potential problems.

The clinical assessment portlets are available through the main ASRB Web portal but, as stated previously, are also deployed on disconnected laptops. Assessments performed on the laptops are uploaded to the central system once the CAO has completed the interview and returned to an Internet-connected office.

3.3 MRI Scanning

The Magnetic Resonance Imaging (MRI) portlets provide the ability for CAOs and MRI technicians to upload, store, and retrieve details of participant scanning sessions, including information such as the date, time and location of a scan, a copy of the scan, whether any artefacts were discovered in the scan, and any comments from the MRI technician.

3.4 Blood Collection

The blood collection portlets allow tracking of blood samples collected from participants, including details such as the location of any samples and the amounts of the various blood products that are available for each participant.

3.5 Data Access

The portlets described above allow the data entered into each section to be accessed, queried, and modified by authorised users. However, the ASRB system also provides read-only access to subsets of the data to any researcher who has been granted approval. To allow such access, a set of portlets exists for querying the system.

Authorised researchers can use the query portlets to ascertain whether the data they require is currently available through the ASRB. This can be done without giving the researcher access to the actual data, i.e. the queries provide tailored meta-data about the actual dataset. For example, a researcher interested in studying how handedness affects schizophrenia can determine if there are sufficient left-handed participants that have completed the clinical assessment process to complete their study, without knowing any of the answers given in any particular assessment.

When a researcher discovers that the data required for a study is available in the ASRB system, the researcher can request access to the data by describing the query that was used. If the researcher's request is authorised by their local Ethics Committee, and by the ASRB Access Committee, a dataset, with either identifiable or deidentified information, is created for the researcher, who can then access that dataset through the ASRB system.

4 SYSTEM EVOLUTION

The general process followed for the improvement of the ASRB IT system is the standard test-driven development process (Beck, 2003). When a new request is sent to the IT team, a new ticket is entered into the ASRB's Trac system (Edgewall Software, 2011), indicating that the request is either a defect that needs to be fixed, a request for an enhancement or new feature, or a task that needs to be performed.

Each individual in the IT team can then view or comment on any of the tickets in the system, and may choose an unassigned ticket to work on. If the request is to perform a task, such as to generate a report or verify some data in the system, then the ticket can be closed once that task is completed.

In the case of a bug fix or feature enhancement, the developer writes a failing unit test that demonstrates the deficiency in the system, and then defines the improvement that allows the test to pass. Once the test passes, the code is reformatted to ensure that it matches the standards required for code in the project. The new test case and improvement are then committed to the ASRB code repository with the comment including "Fixes #n", where *n* is the number of the ticket in the Trac system that is addressed by the new code. This automatically closes the ticket, and the developer can then return to the list of open tickets to find a new issue on which to work.

On every commit to the ASRB code repository, the continuous integration server performs a clean build of the ASRB system. If there are any code violations or test failures or inadequacies, notification is sent to the developer responsible for the changes that caused the problem. The developer must then create a new ticket and resolve the issue using the general development process. While code

that causes such problems should be identified before being submitted to the main repository, the automated notifications ensure the quick resolution of any issues that are mistakenly overlooked at earlier stages. In the case of test failure or inadequacy, the new system build is cancelled before completion so it cannot be accidentally deployed.

5 APPLICATION DEVELOPMENT

The ASRB IT system was first designed in 2006 and a complete remanufacture occurred in 2010. Through that time, the system has constantly been evolving as the needs of the researchers collecting and using the data have changed. The majority of the changes have been feature requests or enhancements, though the most time-critical tasks have typically been the detection and resolution of code defects.

As far as developers of the ASRB are concerned, there is very little difference between a request for a new feature and a request for a feature enhancement. Both involve creating a new unit test for something that the system does not currently do, and then modifying the system so the test passes successfully. Further, the adoption of a test-driven development process means that any code defect is simply seen as an inadequate test suite. Since tests are written before any new functionality is added, and functionality should only be added to ensure that tests pass, there have been relatively few actual defects in the system. One exception to this was with the upload of clinical assessments from the laptops to the server system, as described below.

The server supports different versions of the questionnaire used in the clinical assessment process, and the laptops frequently download the latest version from the server. The unit tests that were written to guarantee the correct upload of the answers of a completed clinical assessment had the incorrect assumption that the server and laptops would always have identical definitions of the questionnaire. When some laptops were using a different version of the questionnaire than the latest version on the server, some answers were stored incorrectly.

Once discovered, the issue was replicated in the test environment and the problem became obvious: a unit test was using a value that is not guaranteed to be consistent across different versions of the questionnaire to identify some responses. The test was completing successfully because it was only using a single version of the questionnaire. A new unit test that used different versions of the questionnaire was created to highlight the incorrect assumption. The code was then modified so that the server could implement a different version of the questionnaire from the laptops, but still successfully receive uploads from those laptops. This stopped any future assessments from being uploaded incorrectly, but some assessments had been uploaded before the problem was discovered, and had incorrect information stored in the database.

Fortunately, the system records, in an audit log, every upload or modification of a participant response. Thus, utilising knowledge of the version of the questionnaire being used by the laptops, and the differences with the server version, it was possible to extract the correct answers for the affected assessments from the audit log. This issue highlights the importance of ensuring that unit tests do not rely on hidden assumptions that are incorrect. To help ensure this, the ASRB development process now requires the explicit declaration of any assumptions made in a unit test, and part of the code review process is to identify any undocumented assumptions.

6 DATABASE MANAGEMENT

Because it stores the data, the database is obviously a critical component of the ASRB system. Apart from some direct access for maintenance purposes as described below, the database is only ever accessed through the Web application. Further, since the Web application only communicates with the database through the Java Persistence API, the complexities of the database are typically hidden from the application developers.

To optimise database performance and ensure data integrity, a script is run nightly. The script begins by creating a backup of the database in a secure location. Once the backup is complete, the script performs some maintenance on the database, including weekly reindexing of all tables and vacuuming of the database as described in the PostgreSQL manual (PostgreSQL Global Development Group, 2011).

The final stage of the nightly script performs tests to verify the correctness of the data and reports on the usage of the system. Data verification ensures that all clinical assessment answers are valid and that any calculated values are correct. These

conditions should be guaranteed by the ASRB Web application, so the nightly checks ensure that any defects with the Web application are discovered quickly. Similarly, the nightly report of usage of the system allows potentially malicious access to be discovered and investigated.

6.1 Direct Access to the Database

While the majority of access to the ASRB database is through the Web application, there have been some tasks for which it has been easier to directly manipulate the database. This includes the migration of clinical assessment data into the system because of a change in data format, the importing of blood information because of an unreliable Internet connection at the storage location, and the bulk upload of MRI scans because the sheer size of the data made off-site uploads unfeasible.

6.1.1 Data Migration

The design of the ASRB database changed slightly when the system was completely remanufactured in 2010. This required the migration of data from the old system to the new format required in the current system. A similar issue occurred when a new site, which had already collected some of the clinical assessment data used in the ASRB in a different format, joined the project. Rather than requiring this data to be re-entered through the new Web application, a set of programs was developed to convert the data to the correct format to allow its use in the ASRB system.

Initially, a naive approach was used when trying to migrate the different data into the current database, with answers being directly imported to the new system from comma separated value (CSV) files. However, this resulted in incorrect data being stored. Some questions, for example, expected the answer to be either “true” or “false”, but the data being migrated stored the answer as “1” or “0”, respectively. A set of programs was created to notify the database manager of any data that was being imported with the incorrect type. The manager could then specify a filter that converted the values stored in the CSV files into the correct format for storage in the database before the import was applied.

Unfortunately, the data migration process could not be automated any more than this. However, once the data was successfully migrated into the system, there was no need to repeat the process. Thus, the semi-automatic method was acceptable. Further, the programs that were developed to check the type of the data in the CSV files are now used as part of the data verification stage of the nightly database maintenance script.

6.1.2 Management of Blood Data

Once the ASRB system was designed and running, it was discovered that the blood technicians at one of the blood storage locations had very limited access to the Internet. Specifically, the computers in their laboratory do not have any Internet access so the technicians must go to a separate office to enter data into the ASRB system. Since the data they are entering is about blood samples processed in the laboratory, this required the technicians to enter the data twice: once into a spreadsheet in the laboratory, and then again into the central ASRB system.

A standard spreadsheet format has been created to prevent the need for double entering of the data. Instead, the blood technicians enter information about samples into the spreadsheet and periodically send a copy to the database manager. The manager has a program that imports data from the spreadsheet into the correct location in the database. In the future, a system similar to the clinical assessment system will be created for the offline entry of blood information. This will allow automated checking of the data entered, which is not possible using the current spreadsheet software.

The limited Internet connectivity of the blood technicians was not discovered until after deployment of the ASRB system. This shows a deficiency in the requirements analysis stage of the development process. Fortunately, in this case, there was an acceptable alternative that managed to avoid the problem. However the circumstance highlights the importance of clearly specifying all dependencies when designing a new system, and ensuring that all such dependencies can be accommodated.

6.1.3 Bulk Upload for MRIs

Magnetic Resonance Images (MRIs) are used by the ASRB to provide a visualisation of the internal detail of the structure of the brains of participants. These images can be over 200Mb in size, and the ASRB MRI technicians often wish to upload a large number of them at the same time. Doing this over an Internet connection, while possible, would be very time consuming. Instead, the images are placed on DVDs or portable hard drives and delivered to the database manager. Once copied onto the database server, a custom program is run that links the image files with scanning sessions entered into the

database. This program is automatic and takes seconds.

This process shows the importance of having different users of an eResearch system cooperate with each other. The copying from disc and running of the program takes the ASRB database manager minutes, most of which are automatic, while the upload through the Web system would take the MRI technicians hours, involving both time consuming Internet uploads, and performance of manual operations to place the various scans in their correct locations.

7 CONCLUSION

The ASRB is an eResearch project that has been running since 2006. The requirements of the system have been evolving since that time, and the technologies and processes used to provide the necessary support have followed this evolution. A test-driven development process has been adopted by the project to allow new features and enhancements to be rapidly added to the system with minimal risk of the new enhancements interfering with existing functionality.

Further steps are taken to help ensure that any defects that are added to the system are discovered quickly. This begins with the continuous integration server, which notifies developers whenever any problems are detected when the code used in the system is changed. Frequent system checks and reports also alert the IT team of any issues with the system. In this way, problems are usually detected when they are small, rather than only being discovered when a catastrophy occurs.

The IT requirements of the ASRB project are continuing to evolve, and the processes used to support such changes must also grow as the system develops. One area of the ASRB development process that needs improvement is that of documentation. The system was originally implemented to be self-documenting, with the issue tracking system, unit tests, and code comments providing valuable information. However, these techniques mainly provide low-level details, and there is need for more high-level documentation. This will increase overall understanding of the system, and will help to minimise confusion and miscommunication between developers.

REFERENCES

Beck, K. (2003). *Test-driven Development by Example*: Addison-Wesley Professional.

Beck, K., & Gamma, E. (2011). JUnit Cookbook Retrieved 12 July 2011, 2011, from http://junit.sourceforge.net/doc/cookbook/cookbook.htm

Biswas, R., & Ort, E. (2006). The Java Persistence API - A Simpler Programming Model for Entity Persistence Retrieved 12 July, 2011, from http://www.oracle.com/technetwork/articles/javaee/jpa-137156.html

Burn, O., Kühne, L., Giles, R., Sukhodolsky, O., Studman, M., & Schneeberger, T. (2010). Checkstyle 5.3 Retrieved 12 July, 2011, from http://checkstyle.sourceforge.net/

Castle, D. J., Jablensky, A., McGrath, J. J., Carr, V. J., Morgan, V., Waterreus, A., et al. (2006). The Diagnostic Interview for Psychoses (DIP): Development, Reliability and Applications. *Psychological Medicine, 36*(01), 69-80.

Collins-Sussman, B., Fitzpatrick, B. W., & Pilato, C. M. (2004). *Version Control with Subversion*: O'Reilly Media, Inc.

Doliner, M. (2006). Cobertura Retrieved 12 July, 2011, from http://cobertura.sourceforge.net/

Edgewall Software (2011). The Trac Project Retrieved 12 July, 2011, from http://trac.edgewall.org/

Henskens, F. A., Loughland, C. M., Aphale, M. S., Paul, D., Richards, J. M., Rasser, P., et al. (2009). *IT Support for the Australian Schizophrenia Research Bank*. Paper presented at the International Conference on Health Informatics (HEALTHINF'09).

Hepper, S., Fischer, P., Hesmer, S., Jacob, R., Taylor, D. S., & McCallister, M. (2005). *Portlets and Apache Portals*. New York: Manning.

Kawaguchi, K. (2011). Meet Jenkins Retrieved 12 July, 2011, from http://wiki.jenkins-ci.org/display/JENKINS/Meet+Jenkins

Mueser, K. T., & McGurk, S. R. (2004). Schizophrenia. *Lancet, 363*(9426), 2063-2072.

Os, J. v., & Kapur, S. (2009). Schizophrenia. *Lancet, 374*, 635-645.

PostgreSQL Global Development Group (2011). PostgreSQL: The World's Most Advanced Open Source Database Retrieved 12 July, 2011, from http://www.postgresql.org/

Sezov, R. (2009). *Liferay Portal Administrator's Guide* (Third Edition ed.): Liferay Press.