VERSI and BMI Project

# 1 Project Overview

Herein we need to think about a variety of stakeholders and a general set of conflicts of interest. Basically, it looks like we need a sales pitch that doesn’t recognise where people are losing.

Some information gathered

Effectively – Neil wants Mediaflux and Daris to succeed – somewhat devout here.

Mediaflux has support/Daris does not.

A whole bunch of other ways to do the same thing – Tardis, XNag.

Likelihood that Monash E-Research are creating something similar for Monash microscopy guys (Tardis we expect) to do exactly the same thing as Daris.

A belief with Anne that Jason is the most flexible person here (albeit with some hesitation on Steve’s behalf to confirm that fact).

Gary Egan is under pressure to deliver. He is not especially easy to deal with right now.

Monash e-research don’t appear to like VERSI very much.

Andrew Lonie, David Abramson are linked but look like they are separating themselves from most of this.

The work is effectively both simple and complex. Simple but it won’t work unless you add something quite complex.

There are problems with the data model.

Mediaflux could work as an ORM replacement…so long as the rest was developed separately.

Interoperability – effectively it is all about schemas.

What are the potentially good bits about this project?

* A potentially good data model for DICOM extensive metadata.
* Integration with computational workflows and their distributive performance.
* Controlled experiment methodologies that can be reused and shared.
* Accessible storage of large data sets.
* Provenance.
* Federated data allows scalability and sharing with multi-site teams.

What are the bad bits?

* Expensive ORM (but if it is considerably better than community versions – it *could* justified)
* Badly Engineered quasi-portal
* A misuse of expressions
* Lack of Researcher input
* Lack of good marketing
* Poor user Interface
* Buggy/unreliable
* Slow performance
* Despotic control of code and development
* CVS used as code sharing tool
* Poor use of a wiki/ difficult to access information

How do we win?

* Researcher uptake!
  + Usability
  + Reasonable flexibility
  + Reliability
  + Good Marketing
* Quality of Code
* Satisfy Monash
* Complete work on time
* NB: Our job is only small – we implement this suspicious looking module called methods. Once done then theoretically ‘All good’.
* We can simply request changes to the underlying layers to work and progress from there.
* My position is difficult due to politics preceding implementation. It looks like Neil went ahead without consulting us and now we have a minor mess.

## 1.1 Project Details

**Domain**: Neuroscience

**Sub-Domain**: MRI, PET, CT

**Type**: Subject-based research

**Subjects**: Animal (subsets), liquid, solid

**People**: Jason Lohrey, Neil Killeen, Gary Egan, David Barnes, Wilson Liu, Andrew Lonie, Simon Milton, David Abfamson

**Organisations**: Monash Biomedical Imaging, Arcitecta, Howard Florey Neuroscience, Centre for NeuroScience – University of Melbourne, Neuroscience Department – Monash University , Neuroimaging and informatics Group.

**Objectives:** Proof-of-Concept over end-user capability. Prove that workflows can be created, adhered to and generate reusable, searchable data for multiple users.[REVISE!!!]

## 1.2 Expressions:

*Workflow*: An external workflow can be executed by a method????

*Nimrod/K*: Nimrod joined with Kepler workflows

*UMLS*: Unified Modelling Language Systems

*OBO*: Open biomedical Ontology

*Foundational Model of Anatomy Ontology*

*Human Disease Ontology:*

*PSS*: Project Object Study. An extension on DICOM to build it into a project hierarchy.

*DICOM*: Digital Imaging and Communications in Medicine. A standard for handling, storing, printing and transmitting information in medical imaging. NB: involves an object model between Patient, visit, study, series.

*CCLRC:* A generic model for handling e-science data

XCEDE: A metadata schema

PSSD: Project/MultiSubject/Multi Method Model defined in the scope of this project.

*Project:* Established by a team to undertake a specifi c investigation.

*Subject:* The subject matter (e.g. animal, plant etc.) of a particular *Project*. There are typically many *Subjects* per *Project.*

*ExMethod:* Container for the execution of a specifi c *Method;* holds reference to *Method* and the state of execution (e.g. executed *Step*) of the *Method*.

*Study:* A container for a class of measurements. For example, a neuroscience study might be of type MR, Microscopy, PET or EEG.

*DataSet:* A set of acquired or processed data that may take any form (e.g. an MR volume)

*State:* The state (changes may be transient or permanent) of the subject at a point in time.

*Method:* The specification of a research process. Methods are applied to *Subject* objects.

* *Integrate computational step into total method*
* *Define metadata*
* *Experimental/computational steps*
* *State change ;;;; eventual connection to another ontologies*

*Step:* A single step in a *Method*. A *Method* may have one or more *Step*s to be performed. *Method*s may allow *Step*s to be performed sequentially or in any order

*State Change:* A specialized *Step* in a *Method* that results in recording a state change for the *Subject*. The state change will be recorded using the metadata specifi ed for the step,

*Data Set Step:* A specialized *Step* in a *Method* that produces one or more *Data Set*s. The *Data Set Step* details the metadata to be generated for the acquired or derived *Data Set*s.

*Branch Step* A conditional branch that refers to one or more other *Method*s. The branch may require one or all of the specified sub-*Method*s be performed.

*R-Subject:* An *R-Subject (R* for *“*re-usable” or “real”) is used when the subject matter participates in multiple *Projects* (e.g. a person).

*MediaFlux*: Digital asset Management. A commercial framework for converting a model to services. Extends existing RIA platforms with libraries for reuse

*DARIS:* An application which connects people to raw data (as assets). Extends media flux services to create a federated query language.

*FEDERATION:* The distribution and mirroring of data across multiple storage servers.

## 1.3 System Roles

1. Administrator
2. Project Manager
3. SubjectAdministrator
4. DataProcessor
5. Developer

## 1.4 High Level Tasks (AS PER GRANT):

1. *Extension of framework*
2. *Importing of file formats*
3. *Integrating MediaFlux Workflows with Nimrod/K, Taverna, etc*
   1. *Service layer*
   2. *Pssd Layer*
4. *Distribution/Federation of data/control across servers*
5. *Query Module for PSSD*
6. *Integrate with Neuroimaging applications*
7. *Extend pssd framework*
8. *Add processing workflows to existing workflows*
9. *A full end-to-end workflow of all types*
   1. *Data acquisition, data ingestion, Quality Control, Notifications, process, analysis*

## 1.5 Systems in Use

1. DARIS
2. MediaFlux
3. DICOM servers --- Imaging Devices
4. GWT
5. XNat
6. Nimrod/Kepler Workflows

## 1.6 Development Lifecycle

## 1.7 Modelling Languages

## 1.8 System Model

### 1.8.1 Component Model

### 1.8.2 Class Diagrams

### 1.8.3 Service Model

### 1.8.4 Modules

## 1.9 Object Model

Basically pssd – reference model on TWIKI

## 1.10 Constraints

## 1.11 Hardware

## 1.12 User Interface

## 1.13 Ongoing Support and Maintenance

## 1.14 Quality Concerns

### 1.14.1 Extensibility

### 1.14.2 Scalability

### 1.14.3 Interoperability

## 1.15 Review & Critique

## 1.16 Success Criteria

# 2. VERSI Implementations

## 2.1. Functions

### 2.1.1 Method Creator

Method

- viewer =>create => monitor

*Can be done*

1. Prototypes to demonstrate a directed graph with related metadata to each step.

2. How do we model a method to a research

3. Documentation/setup wiki/integrate systems

4.

1. More design on object model – collaborative between Jason, Neil and versi.
2. Documentation – specification and Requirements Envisioning (agile driven modelling) [Stored on Global WIKI and Versi’s JIRA]. NB: Iterative.
   1. Iteration 1 [ETA: 7/1/2012 – Iteration 1]. Includes:
      1. Information gathering from meetings, wiki and code
      2. High level modelling of systems
      3. Requirements envisioning for MBI
      4. Research on interoperability and other systems
      5. Definition of existing systems
      6. Design of existing object model
      7. Refinements
      8. Quality guideline
      9. Iteration 1 function points, tasks and user types
      10. High level Use Cases
      11. Details on Constraining Scope.
      12. Sign-off is enough information to complete prototypes
      13. Definition of Modelling Languages
      14. Expected Development Tasks:
          1. Environment setup
          2. Rapid Prototyping
          3. Basic Services
      15. Risk Analysis and Feasibility
   2. Iteration 2: [ETA: 7/2/2012]
      1. Service Layer Definitions
      2. Data Contracts
      3. Object Model Constraints
      4. Confirmed GUI design
      5. Details of Rapid Prototypes
      6. Expected Development Tasks:
         1. View Method
         2. Create Method
            1. Add Existing Step
            2. Create Basic Step
         3. Update Method
         4. Method Dashboard
      7. Low Level Use Cases
      8. Refactoring
   3. Iteration 3: [ETA: 7/3/2012]
      1. As for Iteration 2- parts i-v,vii,viii
      2. Expected Development Tasks
         1. Remove Method
         2. Method Versioning
         3. Method Templating
         4. Method Instantiating
         5. Step Data insertion/update across Method
         6. Shared Steps
         7. Dynamic Steps
   4. Iteration 4: [ETA 7/4/2012]
      1. As for iteration 2 – parts i-v, vii,viii
      2. Expected Development Tasks
         1. Method Searching
         2. Step Searching
         3. Method Tracking
         4. Method Cloning
         5. Multiple Method Selection
3. Documentation: Architectural Design document High level to class structure (nearly SysML) [Stored on Global WIKI and Versi’s JIRA].
   1. Iteration 1 [ETA: 14/1/2012]
      1. System Interfaces
      2. Development Languages
      3. Detailed Object Model
      4. Design Decisions
      5. High Level system Design (SOA/MVC)
      6. Layer Design (N-TIER)
      7. Subsystem structure and links to use cases
      8. System Dependencies
      9. GUI Widgets and language
   2. Iteration 2 [ETA: 14/2/2012]
      1. Class Diagram for subsystems
      2. Software Design Patterns
      3. Collaboration/Sequence Diagrams
      4. State Charts – per task
      5. Applies to all requirements on Iteration 2
   3. Iteration 3 [ETA 14/3/2012]
      1. As per Iteration 2 but applied to related requirements
   4. Iteration 4 [ETA 14/4/2012]
      1. As per Iteration 4 but applied to related requirements
4. Action Plan – low level work items (AON/Gantt - iterative) [Stored on Global WIKI and Versi’s JIRA].
   1. Iteration 1 High Level Tasks [ETA: 22/12/2011]
   2. Risk Assessment for new tasks [ETA:1/1/2012]
   3. Iteration 2 Work Plan [ETA: 1/1/2012]
   4. Risk Assessment for new tasks - 2 [ETA:1/2/2012]
   5. Iteration 3 work plan [ETA: 5/2/2012]
   6. Risk Assessment for new tasks - 3 [ETA:1/3/2012]
   7. Iteration 4 work plan [ETA: 5/3/2012]
5. GUI Prototypes –– GWT expected
   1. Iteration 1 – Task/Use Case based (Wireframes and then interactive interfaces) [ETA: 25/1/2012]
      1. Create/Modify Method
      2. View Method/s
      3. Method/Project Dashboard
      4. Track Method progress
      5. Subject/method metadata creation
      6. View Method Steps and Step Types
      7. Step and Method Sharing
      8. Search for Steps
      9. Search for Methods
      10. Update/Add Data for step
   2. Iteration 2 [ETA: 10/2/2012]
      1. Refactor All
      2. Method Cloning
      3. Add Step Data
      4. Method Versioning
      5. Method Templates
   3. Iteration 3-4 [ETA:30/2/2012]
      1. Refactor All
      2. Shared Steps
      3. Dynamic Steps
      4. Step Plugins
      5. Multiple Method Selection
   4. Refactoring
6. Implemented Object Model (XML/postgreSQL)
   1. Service Layer Calls [ETA: 20/1/2011]
      1. Requests/Responses
      2. Data Generalisation(XML Defined)
      3. Interface Data Model Details – raw service from charts in interface
      4. Refactors
      5. Authentication and ACL
      6. Entity Relationship Description
   2. Integration with pssd/DARIS [ETA: 1/2/2011]
      1. Integration with TCL and MediaFlux
      2. Model Refinement
      3. Model Constraint Engineering
      4. DICOM support
      5. MetaData Extensibility
      6. Entity Relationship description
   3. Interoperability [ETA: 15/3/2011]
      1. Data Map Description (E-R or Z)
      2. XSL Transforms
      3. Service Layer inheritance and extension
      4. Tardis Support
      5. XNat Support
      6. Refactors

### 2.1.2 Doctype/Schema Generator

### 2.1.3 Method Manager

### 2.1.4 Method Selector

### 2.1.5 Step and Workflow Integration

### 2.1.6 Changes to XMethod model

## 2.2 User Interface Requirements

## 2.3 Interoperability adherence

## 2.4 Tools

## 2.5 Process

## 2.6 Development Lifecycle

## Rapid Prototyping

## 2.7 Modelling Languages

### 2.7.1 UML Adherence

## 2.8 System Modelling

### 2.8.1 System Interfaces

### 2.8.2 Component Model

### 2.8.3 Class Diagram

### 2.8.4 Finite state machine

### 2.8.5 Service Model

### 2.8.6 User Interface Design

## 2.9 Documentation

## 2.10 Constraints

## 2.11 Development Timeline

## 2.12 Quality Criteria

### 2.12.1 Performance

### 2.12.2 Extensibility

### 2.12.3 Reuse

### 2.12.4 Standards Adherence

### 2.12.5 Usability

### 2.12.6 Accessibility

### 2.12.7 Maintainability

### 2.12.8 Scalability

## 2.13 Acceptance Criteria

## 2.14 Success Criteria

## 2.15 Work Plan

## 2.16 Design Decisions [LOG]

## 2.17 Ongoing Support

## 2.18 Beyond the scope of Neuroscience