Melding Data Management with Computational Workflows

Neil E B Killeen¹,², Jason M Lohrey³, Michael Farrell⁴, Wilson Liu¹, Slavisa Garic⁵, David Abramson⁵ and Gary Egan⁶

¹ Centre for Neuroscience Research, University of Melbourne, Victoria, Australia
² Information Technology Services, University of Melbourne, Victoria, Australia
³ Arcitecta Pty. Ltd, Victoria, Australia
⁴ Florey Neuroscience Institutes, Melbourne, Victoria, Australia
⁵ Faculty of Information Technology, Monash University, Clayton, 3800, Victoria, Australia
⁶ Faculty of Medicine, Nursing and Health Sciences, Monash University, Clayton, 3800, Victoria, Australia
Overview

- Introduction
- Tools & Technologies used
- Integration of those tools
- New Components
- Case Study and Results
  - Application and Computation results
- Conclusions and future work
Introduction

Generic practical limitations with data handling include:

- Poor integration with acquisition systems
- Poor data management (raw, processed, analysed) and coherent view for all team
- Poor authorisation model
- Separation of data and meta-data management
- Poorly defined quality control steps (downstream pain)
- Poorly defined processing workflows
- Huge amounts of data that can’t be handled manually
Enhanced Approach

Researchers might aim to

- Automate data/meta-data capture
- Hold data in secure repositories
- Integrate capture with upload to repository
- Automate processing via robust workflows and with workflow engines
- Hold raw, processed and analysed data products in the repository
- Integrate workflows with data management
Really?

Not all Analysis Processes Are Created Equally

- Many research processes are ad-hoc and evolve rapidly
- There may be substantial learning value in not using a black box
- Pick and choose components from loosely coupled environment that suit your analysis
Targeting

Class of experiment (e.g. biomedical imaging) where

- large cohorts
- large amounts of data acquired
- standard processes to integrate data capture with repository
- upstream quality control essential
- static and robust workflows applied to all data
- statistical analysis depends on uniform processing
- distributed teams all needing common view
Creating

• An integrated software component environment addressing the above issues
• Modular approach; choose desired components
  – E.g. Full or partial integration with workflows
• Based on mature Technologies
  – Already address some of the deficiencies
  – Mediaflux – Data operating system (Arcitecta)
  – DaRIS/Mediaflux – biomedical imaging informatics framework (University of Melbourne) supplying managed repository
  – Nimrod - Nimrod/G and Nimrod/K (Monash University)
  – Kepler - Scientific Workflow Management System (University of California, San Diego)
Mediaflux
Arcitecta Pty Ltd

- Operating system for data and meta-data
  - Create, manipulate and manage data and meta-data in single or collaborative environments
  - Combines data management with many other services including gelocation, federation, replication, workflow and webserving

- Server is developed in Java
  - Client access by HTTP, NFS, Java, .Net, JavaScript

- Web-based portal for asset collection management
  - All meta-data held as XML and user definable
  - Highly scalable database
Mediaflux

- **Capabilities presented as set of services**
  - Finding, storing & retrieving assets (data+metadata)
  - Archiving and handling large data
  - Data analysis and transformation
  - Extensible plugin platform allows custom packages

- **Strong authorization model**
  - Services for identity management integration
  - Role-based authorisation access to data
  - Each repository has independent access control
Mediaflux
DaRIS Framework
(Centre for Neuroscience Research, University of Melbourne)

Secure repository to manage (bio-medical imaging) data

• Data model – PSSD
  – Applies to studies on a variety of subject types (e.g. animal, minerals)
  – Fully customizable to different research domains (by defining meta-data and Methods)
  – Fine-grained customization on a per Project basis

• Functionality via a Mediaflux package of plugin services
  – Data-model driven interfaces
  – Data uploaded via clients (e.g. DICOM) and portal drag-and-drop

• Web-based portal
  – Google Web Tool kit (with Mediaflux widget library) based
  – Data model driven
    • Full knowledge of the framework
    • No knowledge of domain specific meta-data and methods
Nimrod Tools
(MeSSsAGE Lab - Monash University)

• Nimrod/G (Grid Broker)
  – Parametric experiment execution
  – Distributed scheduling
    • Manages the scheduling across all available resources
    • Can scheduler to meet user defined time or budget constraints
  – Interfaces with Grid Middleware
    • Condor, SGE, PBS, Globus Toolkit
    • Cloud systems – Amazon’s EC2 or Microsoft’s Azure
Nimrod Tools

• Nimrod/K
  – Provides Nimrod/G functionality
    • Built on Kepler’s runtime engine
    • Uses Nimrod/G to run computations on the Grid
  – Scheduler that can be changed dynamically
  – New custom Tagged Dataflow Architecture Director
    • TDA supports concurrent threads of execution in the workflow itself
    • No change required to existing actors to run under TDA Director
Integration

PSSD data model includes a ‘Method’ object

- Describes a set of experimental research steps:
  - Subject creation meta-data
  - Meta-data recording subject state changes (e.g. anaesthesia)
  - May define constant (i.e. pre-defined) and non-constant meta-data
  - State of steps and overall Method execution
  - Data creation steps (e.g. an MR acquisition)
  - Branch points so structure is recursive
  - Steps are not necessarily sequential
  - Not a work flow in the standard definition (WfMC)
Experimental Method

1.4.5-MRI

Anaesthetize → Longitudinal MR → Coronal MR → Recovery → Kill

Step: 1. Anaesthetize
State: {}
Notes: {}

Subject

method: inhalation

induction:
agent: isoflurane
concentration: 3

maintenance:
agent: isoflurane
concentration: 0.5

monitoring: respiratory rate
Integration

Enhance Method to include Transform Step

- Context of how to transform data. E.g. a computational workflow
- Defines
  - Inputs/outputs (and call backs)
  - Optional configuration parameters
  - Transformation definition
- Output of one Transform Step may be used as an input to another Transform Step
- Transformation dispatched by Transform Service Provider
New Components

• **Mediaflux Transform Framework**
  – **Transform Provider**
    • entity to execute transform
    • controls (e.g. suspend/resume) and monitor
  – **Transform Definition**
    • includes Transform Provider
    • definition of external transform (workflow)
    • parameters required to run it
  – **Transform Instance**
    • actual instance of the transform in Mediaflux
    • current state of the execution
    • parameter values
  – **Transform Output**
    • Outputs stored as assets in Mediaflux
    • Call backs to manipulate outputs (e.g. Q/C)
New Components

- **PSSD Data Model**
  - Enhance to include new Transform Step
    - References the Transform Definition (how to execute external process)
    - Select inputs (PSSD DataSets) for the Transform Step
      - Makes use of query language
    - Specifies outputs (PSSD DataSets) and meta-data describing transform and labelling processed data
    - Call backs for post-processing
    - On execution, the appropriate Transform Definition is submitted for execution via the Transform Framework.
  - Virtual Subjects allows multi-Subject workflows
  - Service layer enhancements
  - Tight or loose coupling to workflows
New Components

• Nimrod/K
  – Kepler implementation of the Transform Service Provider
  – Remote control of the Workflow Execution
    • Start, Suspend, Resume and Terminate
  – Mediaflux Actors
    • Retrieve assets from a Mediaflux repository
    • Create and store new assets to a Mediaflux repository
    • Generic actor that can execute any service
  – Improvements to the Nimrod/K core
    • Nimrod/K Director – Execution control
    • GridJob Actor – Directory transfers, file filtering, placeholder substitution
Case Study - Imaging Clinical Pain

- Identify brain activation associated with lower back pain in people with musculo-skeletal disease
  - fMRI for acquired for two groups of participants (control/patient)
  - Data acquired consisted of
    - Structural brain image
    - Functional brain images acquired every 6 seconds
    - Participants pain rating acquired every minute
  - Identical analysis of data for all participants
    - Derive regional cerebral blood flow from fMRI
    - Statistical analysis to derive brain regions uniquely correlated with pain
• MR data and pain ratings were all acquired at the Royal Children’s Hospital
• DICOM data were uploaded directly from RCH to UM DaRIS instance (with meta-data harvesting) via standard DICOM client
• DaRIS DICOM engine integrated with PSSD data model
• The pain ratings (per Subject) were uploaded via the DaRIS portal (drag-and-drop) into their own PSSD DataSet.
Case Study – Execution

• Standard processing tools are
  – FSL
  – SPM/Matlab
  – Mixture of GUI use and scripted elements
• Developed scriptable elements for all processing components
• Created a Kepler Workflow using these elements
  – Input data sourced from DaRIS by the workflow
  – Output data uploaded back to DaRIS DataSets by the workflow
    • Usually large directory structures in an archive
• Nimrod/K runs the workflow with user specified parameters
• Virtual subject, separate workflow for group analysis
Case Study – Execution

Nimrod/K Director

Get Pain Activations

Get FEAT Design

FEAT: Generate Ensemble Statistics

Store Ensemble Statistics
Case Study – Results

Results show that low back pain participants have
- Decreased levels of blood flow in subgenual cingulate cortex
- Increased levels of blood flow in left inferior frontal gyrus
Case Study
Computational Results

![Graph showing computational results](image)
We have enhanced and integrated DaRIS/Mediaflux and Nimrod/Kepler so that computational work flows can be executed:

– Infrastructure Developed
  • Generic Mediaflux Transform Framework
  • DaRIS/PSSD Transform Step using this framework
  • Nimrod/Kepler implementation of Transform Provider with Mediaflux/DaRIS integration actors

– Modular approach so users can
  • Work with all integrated components
  • Operate workflow from DaRIS/Mediaflux or Kepler
  • Independently process data but still upload and describe it
Future Work/Challenges

• Inconsistently labelled data in the repository; fuzzy logic
• DaRIS Portal
  – How to present/explore/access raw & processed data
  – New components to enable the easy creation and execution of Methods with Transform Steps
  – Interactive Transform Steps (e.g. Q/C)
• Distributed identity management/authorisation
  – Mediaflux as a secure wallet
  – Operating in grid
• Nimrod/K
  – running in server mode
  – New data transport mechanisms
• Making the end-to-end process straightforward 😊
Acknowledgements

- This work was funded by an ARC Linkage grant