Software Automation in Scientific Research Organizations

Dr. Mark Vigder
Institute for Information Technology
• **Managing the software**
  – Science cannot be separated from the software used to support it
  – integration and interoperability, controlled evolution, configuration management, distribution and deployment

• **Managing the data**
  – Everything depends on the integrity of the data
  – Provenance, lineage, distribution

• **Managing the scientific process**
  – What are the activities undertaken in the discovery of new knowledge?
  – Processes must be repeatable - results must be reproducible
  – Managing may include: capture, document, archive, share, execute and reproduce processes
Institute for Aerospace Research

- **NRC-IOT:** support of Canada's ocean technology industries
  - Offshore engineering basin
  - Towing tank
  - Ice tank
  - Cavitation tunnel

- **NRC-IAR:** R&D related to the design, manufacture, performance, use, and safety of air and space vehicles.
  - Wind tunnels
  - Jet engine testing
  - Chicken guns
Hibernia
Yacht
CF700 turbofan, data fusion development
Framework for Scientists

- **Objective:** build a software framework that supports some of the activities and processes of a research organization:
  - Software tool integration
  - Workflow specification
  - Information management
- **An 'IDE' for scientific software tools**
Goals of the Framework

- **Integration of the off-the-shelf software tools**
  - Common means for accessing different tools
  - Integration and interoperability between applications

- **Automate the workflows of the organization**
  - Identify and automate the standard activities and processes
  - Customization of the processes as needed

- **Technology refresh**
  - but bring along the legacy software

- **Improved data and information management**
  - Managing data: archiving, searching, provenance
  - Full configuration management
  - Reproducible
Software related activities

• **Data analysis activities**
  – Gathering, cleaning, transforming, reporting
  – Knowledge discovery from data

• **Tool integration**
  – Programmatic interfaces
  – Wrapping, adapting and extending applications
  – Data transfer between applications

• **Workflow specification**
  – Organize and execute the data analysis tasks in the proper sequence
  – Develop processes organizations use during analysis

• **Information and data management**
  – Archiving data, managing the archived data
  – Configuration Management
Organizational activities

Low scientific domain knowledge → High software development knowledge

High scientific domain knowledge → Low software development knowledge

Tool Integration

Workflow Specification

Information Management

Data Analysis, Workflow activation
Scientific Workflows

- **Workflow**: Movement of documents and/or tasks through a work process (Wikipedia)
  - Structuring of tasks
  - Order and synchronization of tasks
  - Well established in business (BPEL)

- **Scientific workflows**
  - Throughput of data through various algorithms, applications and services
  - Use of multiple interconnected tools
  - Use of multiple data formats
Example workflow: Regular wave calibration

- Load Data
- Data Cleansing
- Variance Spectral Density Analysis
- Zero Crossing Analysis
- Segment Selection
- Zero Crossing Analysis
- Present Results
Start

Safety
- Band Pass Filter
  - Trend, Extrapolate
  - Degradation Model

Stability
- Moving average
  - Differencing

Modeling
- Segment Selection
  - FFT, DTFT Analysis
  - Report Generation

Report Summary
End-user

• End-users are the scientists and technicians
• Many cannot (or will not) write any software
• Must be able to:
  – Find, select, configure workflows
  – Explore the data using different software tools
  – Manage the data generated, software versions used
• For the end-user:
  – Static variation points, presented in a dynamically generated GUI
  – Tools for entering data structures in variation points
  – Organizing and navigating workflow invocations
Workflow developer

• Some software development knowledge required
• Accessible to knowledgeable end-users
  – Written as simple scripts that invoke services
  – Full programming language capability (Python)
  – Tools gathered into a 'toolbox' and dynamically linked to the services are available to workflow developer
  – Workflows are represented as parameterized templates where the parameters represent the variation points
  – Metadata is used to describe the templates and dynamically build the GUI
• Develop wrappers for the tools
• **Warning: Nerds at work!**
  – Tools dynamically added to toolbox - can be customized to a domain
  – Parameter types understood by end-user - custom widgets can be added
  – Data management utilities
  – Logging, exception handling, other utilities...
Observations

- Dynamically generated GUI’s very successful.
- A great deal of the success was due to the strong software engineering group within IOT. IAR is more challenging.
- Many of the standard software engineering techniques were introduced into the organization and were quickly adopted.
- Quick payback by automating easy tasks.
• Questions?
Representing workflows

- **Automation**
  - Represent the workflow in an executable form
  - Engine for executing workflow
  - Invocation of software tools
  - Integration and interoperability of software

- **Re-usability**
  - Similar processes used in many experiments
  - Ease of customization by domain experts
  - Repository for storing, retrieving and managing workflow representations

- **Ease of creation**
  - Minimal software programming knowledge required
def do(self,
    dac_file_name = '',
    target_wave_height = 0.0,
    target_wave_period = 0.0,
    analysis_segment_start_time = 0.0,
    analysis_segment_end_time = 0.0,
    best_cycles_segment_start_time = 0.0,
    best_cycles_segment_end_time = 0.0):

    ""
    * dac_file_name = DAC File Name
    * target_wave_height = Target Wave Height (m)
    * target_wave_period = Target Wave Period (s)
    * analysis_segment_start_time = Analysis Segment Start Time (s)
    * analysis_segment_end_time = Analysis Segment End Time (s)
    * best_cycles_segment_start_time = Best Cycles Segment Start Time (s)
    * best_cycles_segment_end_time = Best Cycles Segment End Time (s)
    """
Managing RunSets and Runs

Workflow template

Extract Metadata

Parameters (RunSet)

Configure

RunSet

Configure

Parameters (Run)

Configure

Workflow
Sweet Design