Generating eScience Workflows from Statistical Analysis of Prior Data

Article · January 2005

Source: OAI

Jane Hunter
The University of Queensland

Generating eScience Workflows from Statistical Analysis of Prior Data

Jane Hunter[1] and Kwok Cheung[2]

[1] DSTC, Brisbane, Australia 4072
jane@dstc.edu.au
[2] ITEE, The University of Qld, Brisbane, Australia 4072
kwokc@dstc.edu.au

Abstract. A number of workflow design tools have been developed specifically to enable easy graphical specification of workflows that ensure systematic scientific data capture and analysis and precise provenance information. We believe that an important component that is missing from these existing workflow specification and enactment systems is integration with tools that enable prior detailed analysis of the existing data – and in particular statistical analysis. By thoroughly analyzing the existing relevant datasets first, it is possible to determine precisely where the existing data is sparse or insufficient and what further experimentation is required. Introducing statistical analysis to experimental design will reduce duplication and costs associated with fruitless experimentation and maximize opportunities for scientific breakthroughs. In this paper we describe a workflow specification system that we have developed for a particular eScience application (fuel cell optimization). Experimental workflow instances are generated as a result of detailed statistical analysis and interactive exploration of the existing datasets. This is carried out through a graphical data exploration interface that integrates the widely-used open source statistical analysis software package, R, as a web service.

1 Introduction

The recent proliferation of eScience communities has led to a demand for more sophisticated information technologies that can assist users to seamlessly capture and interpret experimental data in order to prove or disprove a particular scientific theory or hypothesis. In particular, workflow technologies represent an increasingly important component of the scientific process. They enable eScientists to describe and carry out their experimental processes in a repeatable and verifiable way. Consequently there are a number of international research groups that are concentrating on developing workflow specification and enactment systems that enable scientists to easily define, save, edit, share and re-use their workflows. The more recent systems are based on BPEL4WS (Business Processing Language for Web Services) [1] and graphical interfaces that enable users to combine and orchestrate a number of Web Services (both local and remote) in order to carry out a higher-level complex scientific task or experimental process. This web services-based approach also fits neatly with the refactoring of OGSI (Open Grid Services Infrastructure) [2] into WSRF (Web Services Resource Framework) [3].

Given the data and metadata that is captured through pre-defined workflows, scientists use a variety of methods to process, analyse and interpret the data in order to prove or disprove particular theories or hypotheses. The problem with current eScience workflows, is that they only support the chaining of steps leading up to scientific knowledge discovery. The knowledge extracted from the final data analysis and interpretation phase is often in a form that cannot easily or automatically be re-cycled back to inform the design of new experimental workflows and experimental data capture. This capability is increasingly important as eScience becomes more collaborative and distributed, relying on geographically-distributed groups of scientists working together to capture, share, correlate and analyse large-scale data sets in order to solve complex problems. Figure 1 below illustrates the eScience knowledge cycle that we aim to facilitate.

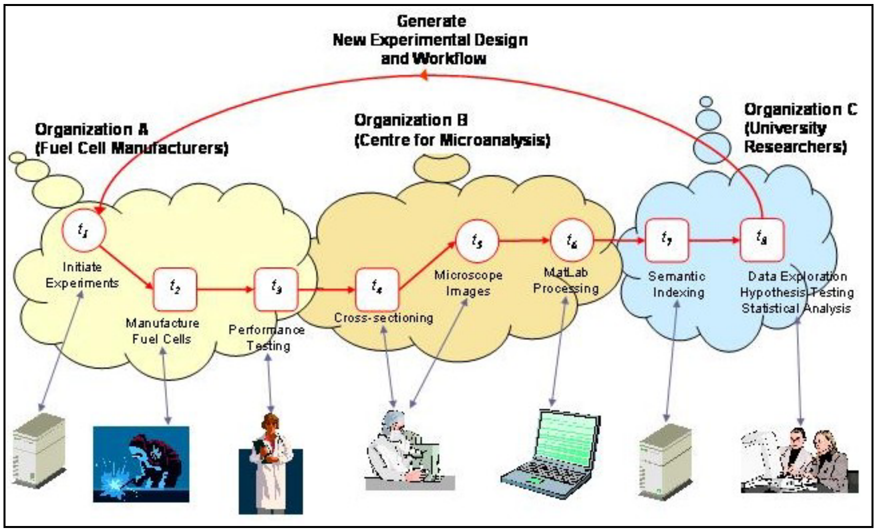


Figure 1 : A Collaborative eScience Workflow Involving Multiple Organizations

The research that we describe in this paper, focuses on the development of an integrated system called FUSION [4], that combines workflow specification and enactment with an interactive data exploration and statistical analysis interface. Although we have developed FUSION for a particular eScience application (the optimization of fuel cells by fuel cell experts) the research described here is applicable across any domains (e.g., bioinformatics, engineering, homeland security, social sciences) that are attempting to solve complex problems through the analysis and assimilation of large-scale, mixed information and data sets.

The remainder of this paper is structured as follows. The next section describes the fuel cell problem scenario in more detail. Section 3 describes related work and objectives. Section 4 describes the architectural design of the system and the motivation for design decisions that were made. Sections 4 and 5 describe the interactive data exploration, statistical analysis and hypothesis formulation interfaces respectively. Section 6 describes the results of evaluating the system on real fuel cell data and images. Section 7 contains concluding remarks and plans for future work.

2. Fuel Cell Optimization – an eScience Example Scenario

Fuel cells offer an alternative, clean, reliable source of energy for residential use, transport and remote communities. Their efficiency is dependent on the internal structure of the fuel cell layers and the interfaces between them. The manufacturing process for fuel cells is extremely complex and we will not attempt to describe it here. Figure 2 illustrates the steps involved in manufacturing and testing just the electrolyte component of a single fuel cell. Associated with each step are a set of parameters that are documented in the database. Only some of these parameters are controllable and the overall objective is to determine the optimum settings for these controllable parameters in order to minimize the cost and maximize the strength, efficiency and longevity of the fuel cell.

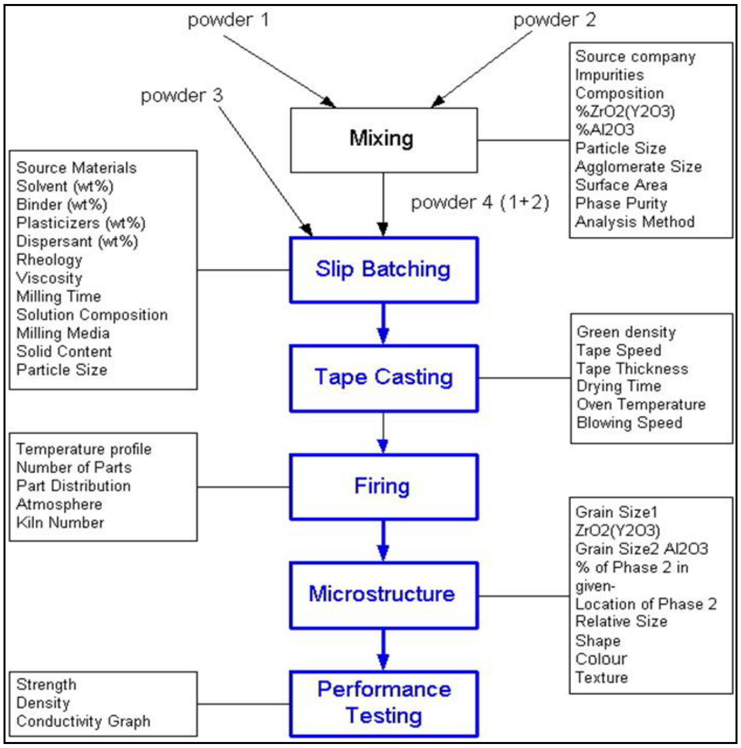


Figure 2: Manufacturing and Analysis Workflow for a Fuel Cell Electrolyte

Electron microscopy generates images of cross-sectional samples through fuel-cell components that reveal complex multi-level information. Simple macro-level information such as the thickness of the cell layers, surface area, roughness and densities can be used to determine gas permeation of the electrode materials. Nano-level information about the electrode's internal interface structure provides data on the efficiency of exchange reactions. Figure 3 illustrates the range of image data obtainable.

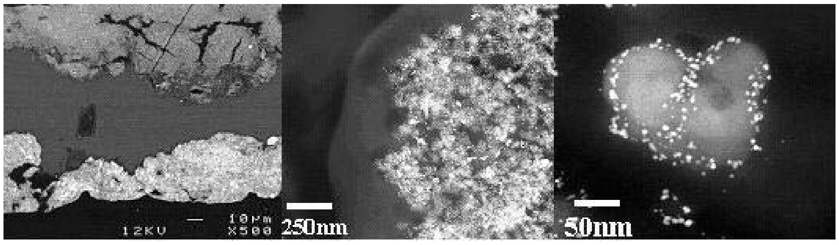


Figure 3: Microscopic images of a fuel cell at 3 different magnifications

By digitising the images and applying image processing techniques (MATLAB [5]) to them, the amount of information expands even further to levels where human processing is not possible and more sophisticated means of data mining are required. In addition to the microstructural information revealed by the images, there are the manufacturing conditions and processing parameters used to produce the cell configurations. Finally, for each cell configuration, performance data is available and the crux of the project is to marry the microstructural data with manufacturing and performance data to reveal trends or relationships which could lead to improvements in fuel cell design and efficiency. Table 1 shows the range of parameters we are dealing with, in addition to the fuel cell images captured at different magnifications.

Table 1. Fuel Cell Parameters

Manufacturing	Performance	Fuel cell characteristics
Wt% Y2O3 - ZrO2	Strength	Layer thickness
Wt% Al2O3	Density	Composition
Wt% Solvent	Conductivity graph	Density
Solid Content	Efficiency	Particle Size and Shape
Viscosity	Lifetime	Nearest neighbors
Tape Speed and Thickness		Surface area
Drying Temperature and Time		Porosity
Cost		Surface roughness

The aim of the work described here was to develop a system that will provide the data and metadata capture and analysis services required for the workflow illustrated in Figure 2. We also needed to build and test an interactive interface that enables fuel cell experts to quickly and easily explore the fuel cell images and data in order to determine associations or patterns between parameters, formulate and validate hypotheses. In addition the user interface must enable scientists to invoke R packages [6] as web services that can be applied to the data to determine where more data is required. From this, new experimental workflow instances (based on BPEL4WS) will be generated and enacted via the BPWS4J engine [7].

3 Related Work and Objectives

3.1 Existing Workflow Languages, Design Tools and Enactors

A number of research groups have been investigating workflows specifically for E-Science and that support the needs of collaborating groups of distributed scientists. Although significant progress has been made in developing workflows for eBusiness, the requirements for eScience workflows are very different [8,9]. Scientists need to be able to store, share and publish their workflows. eScience workflows are more dynamic than business transactions and need to be easily adapted on-the-fly. They need to support large volumes of unstructured data flowing across multiple organizations. The three most significant existing open source workflow systems, designed specifically to support eScience, SCUFL, Kepler and YAWL are described below. Other workflow engines that we have not considered here include TRIANA and ICENI.

SCUFL (Simple Conceptual Unified Flow Language) was developed as part of the ^{my}Grid [10] project – a UK EPSRC-funded project launched in collaboration with the European Bioinformatics Institute and the Human Genome Mapping Project in 2002. Within the ^{my}Grid project, researchers have developed a graphical toolset and workflow enactor that uses its own high level representation of a process flow, including specification of processing units, data transfer and execution constraints. The two main sub-components [11] of the ^{my}Grid project are:

- the Taverna subproject's workflow language called SCUFL (Simple Conceptual Unified Flow Language) and the SCUFL workbench - software tools to facilitate easy design of workflows over distributed computing resources by e-Science communities (e.g., a graphical workflow editor) [12].
- the Freefluo subproject - Freefluo is an engine for enacting workflows composed from web services, especially workflows composed by the SCUFL workbench - which converts a graphical workflow into a business process written in SCUFL. Currently beta-06 version of Freefluo, which was released in September 2003, is available [13].

Freefluo does not support a number of required flow model elements such as: Transition Conditions on Control Links, Join Condition on an activity, Exit Conditions on an Activity and UDDI Query by String. The lack of transition conditions effectively removes support for choosing a workflow branch based on actual runtime data. This is a necessary requirement for the workflow system for this project.

Kepler [14] is another extensible workflow system aimed at scientific workflows. The Kepler project is cross-project collaboration between SDM (Scientific Data Management) Center, SEEK (Science Environment for Ecological Knowledge), GEON (Cyberinfrastructure for the Geosciences) and RoadNet (Real-time Observatories, Applications, and Data Management Network). The aim of Kepler is to provide a framework for design, execution and deployment of scientific workflows. Kepler is built on top of Ptolemy II [15] – an API for heterogeneous, concurrent modeling and design. Kepler currently provides the following major features [16]:

- Prototyping workflows: Kepler allows scientists to prototype scientific workflows before implementing the actual code needed for executions
- MoML – an internal XML language for specifying component-based models and composing actors into workflows
- Distributed execution (Web and Grid-Services): Kepler's Web and Grid service actors allow scientists to utilise computational resources on the network in a distributed scientific workflow.
- Database access and querying: Kepler includes database interactions.
- Other execution environments: Support for foreign language interfaces via the Java Native Interface provides the flexibility to reuse existing analysis components and to target appropriate computational tools.

Kepler was considered as a possible candidate for building our workflow management system. However, at the start of this project, Kepler's source code was not available and the timing of its availability was uncertain – it has since been made available as open source.

YAWL [17] (Yet Another Workflow Language) is a workflow language based on a rigorous analysis of existing workflow management systems and workflow languages [18]. Even though YAWL is based on Petri nets, it has been extended with features to facilitate patterns involving multiple instances, advanced synchronisation patterns, and cancellation patterns. Moreover, YAWL allows for hierarchical decomposition and handles arbitrarily complex data. Workflow specifications are defined from three key perspectives in YAWL:

- Control-flow perspective: The control-flow perspective of YAWL focuses on the ordering of tasks. There are three features offered from this perspective by YAWL: the OR-join task, multiple instances of a task (atomic or composite) and the remove tokens task (i.e. cancellation of a region);
- Data perspective: The data elements could be defined and used for conditional routing, for the creation of multiple instances or for exchanging information with the environment.
- Operational perspective: The operational perspective describes the elementary actions executed by tasks, where the actions map into underlying applications.
- Like Kepler, YAWL is also an open-source project and was regarded as one of candidates for our project.

Unfortunately, all three potential workflow management systems – Freefluo, Kepler and YAWL were unsuitable for use within this project. Freefluo does not support Transition Conditions on Control Links. This is necessary for the workflow implemented in this project, so that the workflow engine can determine the invocation of a particular web service under a predefined condition. Kepler could be an appropriate application for supporting the workflow implemented in this project, but its source and binary codes were only released recently - too late for use in this project. Similarly the source and binary codes for YAWL were only released recently. YAWL also appears to be inadequate for this project because it currently does not support the operational perspective. This means that it does not support web services invocation which is essential for this project.

3.2 Our Objectives

Because the workflow we are considering (Figure 2) is fixed, the graphical workflow design tools are not of high priority and can be implemented later by re-using one of the open source solutions described above. Our primary objective was to enable BPEL4WS workflow instances to be generated and enacted – as a result of interactive and statistical analysis of existing data sets. This aspect is original and not currently supported within any of the existing workflow systems described above. Our second objective was to evaluate this approach by building a demonstrator for a particular application domain and group of collaborating and distributed scientists. For this we have chosen the fuel cell optimization domain because it is a typical scientific problem involving a large number of variables, highly heterogeneous data, multiple organizations and complex processing.

4 System Architecture

Figure 4 illustrates the overall architecture and major components of the FUSION system. There are six major system components:

- Apache Tomcat;
- MySQL Database server;
- MATLAB;
- R statistical analysis packages;

- the Mail User Agent;
- The Data Exploration and Hypothesis Testing Interface.

4.1 Apache Tomcat

Apache Tomcat [19] is the servlet container used in the official Reference Implementation of the Java Servlet and JavaServer Page technologies. Three web applications are installed on Apache Tomcat. The first one is BPWS4J – a workflow engine for deploying and executing business processes written in BPEL4WS. The second one is a web application which acts as the interface between users and BPWS4J. The last one is Apache Axis [20]. Apache Axis is essentially a SOAP engine – a framework for constructing SOAP processors such as clients, servers, gateways, etc. The main usage of Apache Axis is as a web-service container. Four web services are maintained in Apache Axis. The first one is used for retrieving data from, and storing data to, the MySQL relational database running on a database server. The second one is used for sending e-mail messages. The third one uses MATLAB algorithms to extract region data associated with fuel cell microstructure from the cross-sectional fuel cell images. The fourth one invokes R packages that apply various statistical analyses to the fuel cell data.

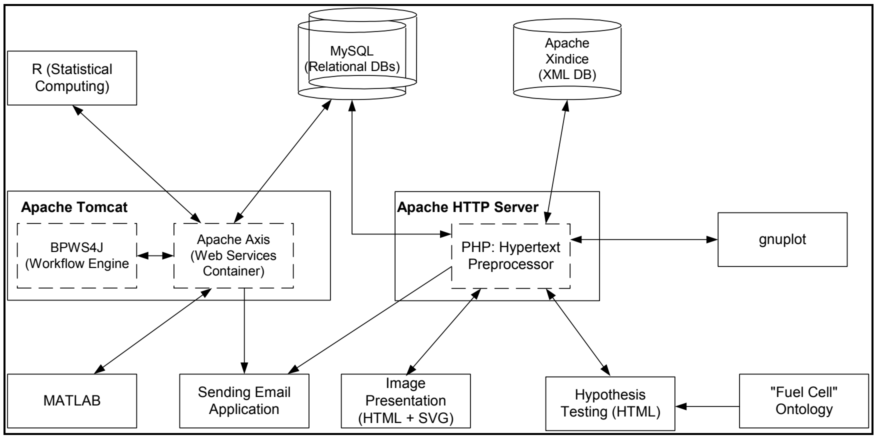

R (Statistical Computing)
MySQL (Relational DBs)
Apache Xindice (XML DB)
Apache Tomcat
BPWS4J (Workflow Engine
Apache Axis (Web Services Container)
Apache HTTP Server
PHP: Hypertext Preprocessor
gnuplot
MATLAB
Sending Email Application
Image Presentation (HTML + SVG)
Hypothesis Testing (HTML)
"Fuel Cell" Ontology

Figure 4: Apache Tomcat-based Web Services Architecture

4.2 MySQL Database Server

Five separate MySQL [21] databases were set up for each task in the *electrolyte manufacture* process (see Figure 2). This is to simulate the situation in which each of the tasks is remotely executed at different locations. In addition, two more databases were created; one for administrative purposes and the other for storing the data generated by the experimental design stage – which generates the instances of the *electrolyte manufacture* process.

4.3 MATLAB

MATLAB [5] is a software package for numerical computation. It began as a "MATrix LABoratory" program, intended to provide interactive access to libraries of state-of-the-art numerical routines. These are carefully tested, high-quality general-use packages for solving linear equations and eigenvalue problems. The goal of MATLAB was to enable scientists to use matrix-based techniques to solve problems, without having to write programs in traditional languages like C and FORTRAN. More capabilities have been added as time has passed, in particular high quality graphics and interface support, the Simulink package for simulating dynamical systems with a graphical interface/ block diagram scheme, and several additional "Toolboxes" (collections of additional specialized functions). In this project, MATLAB is used for extracting microstructural data about fuel cells from the regions in cross-sectional images.

4.4 R Software for Statistical Computing

R [6] is an open source implementation of the S language and environment developed by Bell Laboratories for statistical computing and graphics. R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, etc.) and graphical techniques for analyzing data, and is extensible via *packages*. There are about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics.

Our objective here is to enable the relevant R statistical packages to be integrated into the data exploration interface to determine where more data is required and generate new experimental workflow instances. This is implemented by making the relevant R packages available as web services that can be invoked and applied to the experimental data. In particular, the fuel cell scientists use R for linear modeling of relationships between parameters and would like to highlight areas where more data is required in order to properly validate a theory or hypothesis. Given the ranges of controllable parameters for which more results are required, the system will generate and initiate new experimental workflow instances and notify the relevant agents or individuals.

4.5 Mail User Agent

Mail User Agent is a Java application built on top of JavaMail API [21] which provides a platform-independent and protocol-independent framework in which to build mail and messaging applications. In this project, e-mail is used as a means of notifying users when they have been allocated particular tasks as part of experimental workflows.

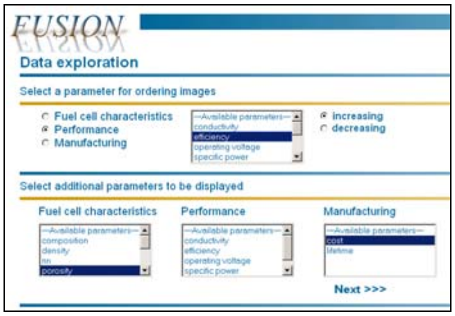
4.6 The Data Exploration and Hypothesis Testing Interface

A browser-based interface was developed to enable the fuel cell scientists to correlate and explore the data through multimedia presentations. Access to the multiple distributed MySQL repositories is through a Web browser (Microsoft Internet Explorer) and an ODBC interface. The Microsoft implementation of SMIL, HTML+TIME [22], is used to build the multimedia presentations. It allows spatio-temporal relationships between information objects as well as visual effects (such as fading between images) to be implemented. Dependencies between values are represented graphically using SVG [23]. Presentations and graphs are dynamically generated using Python scripts. The HTML forms and pull-down menus presented to the user are generated from domain-specific XML schemas and OWL[24] ontologies, described in earlier work [25] and which are specified during system configuration. This architecture is extremely flexible and can quickly and easily be adapted to other domains by connecting to different backend schemas and ontologies. Users can choose the aspects of the fuel cell data which they are interested in by selecting the parameters from the data set to be viewed. For example, *porosity*, *efficiency* and *cost*. The HTML interface also allows users to specify preferences for displaying the retrieved results. Users can specify the following display preferences:

- an ordering parameter for structuring and presenting the results;
- selection of any additional parameters to be displayed;

- the type of presentation mode required (time-based or static);
- preferred data presentation formats (values displayed graphically, or in a list);
- any special effects to be applied to the presentation (e.g., fading etc.).

Figure 5 shows the user interface in which the user has specified that they wish to "order retrieved fuel cell data by increasing efficiency" and "also display values for porosity and cost". The results of the query are processed and transformed into the necessary format by the Presentation Generator which makes decisions about the spatio-temporal layout of the result sets based on the users' preferred presentation mode, format and any special effects. Figure 6 shows the user interface for specifying presentation preferences. There are three possible presentation modes to choose from: a time-based mode (slide-show) or one of two possible static modes – interactive and thumbnail views. The slide-show mode also displays an animated graph together with the images and any additional parameter values chosen by the user. The SVG graph plots one or more parameter values against time, and is generated dynamically in synchronization with the fuel cell images. This enables users to relate visual features in the images to manufacturing and/or performance data. In addition, fading effects can be applied to the images in the slide-show. This is helpful for distinguishing differences across sequential images. Figure 7 illustrates the results of a slide-show presentation. The data exploration interface has been designed to enable a user to interactively explore large mixed-media, multidimensional data and information sets. By enabling users to choose the presentation style which best suits them, and to focus on the range and scope of data sets that most interest them, the system maximizes the potential and speed at which domain experts can discover new interdependencies or trends within the data or develop hypotheses. If the user finds an interesting pattern or association they can save the HTML+TIME+SVG presentation, together with the associated metadata (Unique ID, Date/Time, Creator, Settings, Objective) or move on to the next stage, the hypothesis testing interface.



FUSION

Data exploration

Select a parameter for ordering images

○ Fuel cell characteristics
● Performance
○ Manufacturing

--Available parameters--
conductivity
efficiency
operating voltage
specific power

● increasing
○ decreasing

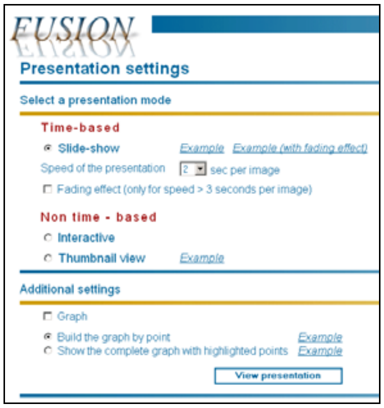
Select additional parameters to be displayed

Fuel cell characteristics
--Available parameters--
composition
density
nn
porosity

Performance
--Available parameters--
conductivity
efficiency
operating voltage
specific power

Manufacturing
--Available parameters--
cost
lifetime

Figure 5. Data Organization



FUSION

Presentation settings

Select a presentation mode

Time-based

● Slide-show   Example   Example (with fading effect)

Speed of the presentation 2 sec per image

□ Fading effect (only for speed > 3 seconds per image)

Non time - based

○ Interactive

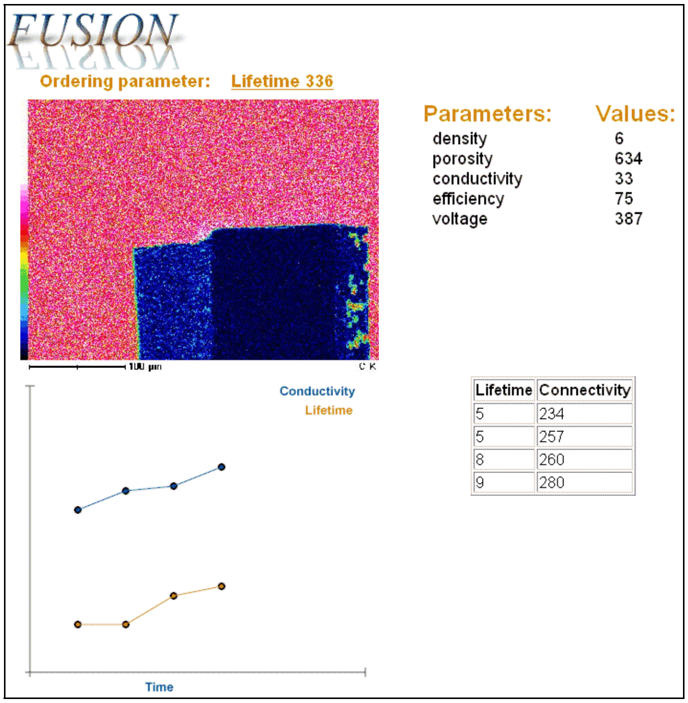
○ Thumbnail view   Example

Additional settings

□ Graph

● Build the graph by point   Example
○ Show the complete graph with highlighted points   Example

Figure 6. Presentation Setting



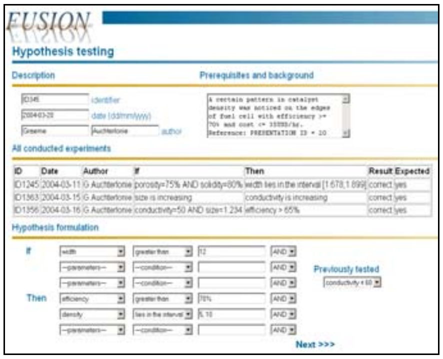
Ordering parameter: Lifetime 336

Parameters:	Values:
density	6
porosity	634
conductivity	33
efficiency	75
voltage	387

Lifetime	Connectivity
5	234
5	257
8	260
9	280

Figure 7. Screenshot of a slide-show presentation with animated graphs

Consider the following example. The user wants to test the hypothesis: "IF substrate width is greater than 12μm AND density value lies between 5–10 particles/μm^2 THEN efficiency is greater than or equals 80%". Figure 8 illustrates the HTML interface that was developed that enables the user to specify such a hypothesis.



Hypothesis testing

Description

ID345 identifier

2004-03-20 date (dd/mm/yyyy)

Greeme Auchterlonie author

Prerequisites and background

A certain pattern in catalyst density was noticed on the edges of fuel cell with efficiency >= 70% and cost <= 35555/hr. Reference: PRESENTATION ID = 10

All conducted experiments

ID	Date	Author	If	Then	Result	Expected
ID1245	2004-03-11	G Auchterlonie	porosity=75% AND solidity=80%	width lies in the interval [1.678;1.899]	correct	yes
ID1363	2004-03-15	G Auchterlonie	size is increasing	conductivity is increasing	correct	yes
ID1356	2004-03-16	G Auchterlonie	conductivity=50 AND size=1.234	efficiency > 65%	correct	yes

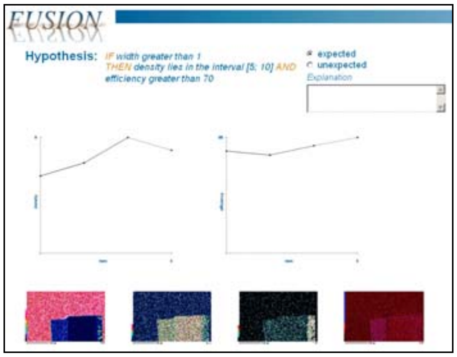
Hypothesis formulation

If	width	greater than	12	AND
	--parameters--	--condition--		AND
	--parameters--	--condition--		AND
Then	efficiency	greater than	70%	AND
	density	lies in the interval	5, 10	AND
	--parameters--	--condition--		AND

Previously tested: conductivity < 60

Next >>>

Figure 8. Hypothesis specification



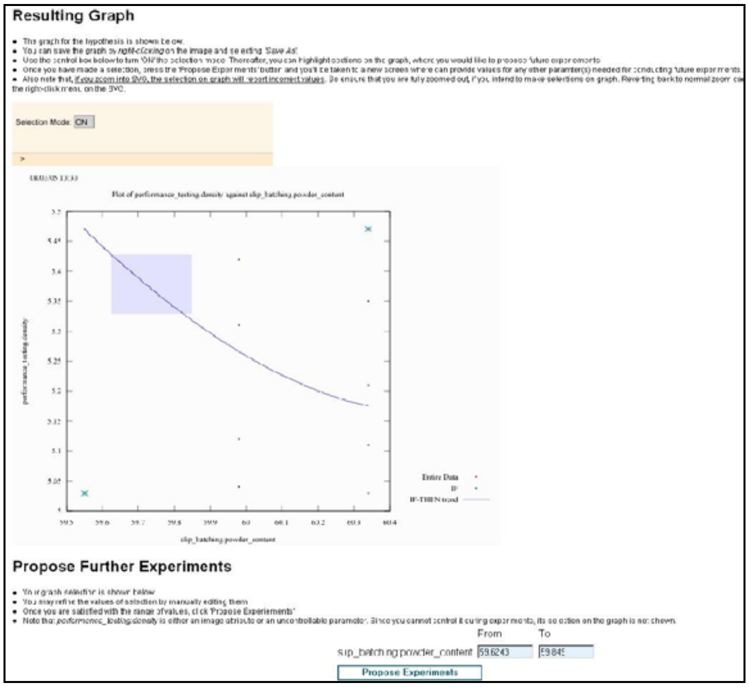
Hypothesis: IF width greater than 1 THEN density lies in the interval [5; 10] AND efficiency greater than 70

● expected
○ unexpected

Explanation

Figure 9. Results of Hypothesis testing

Figure 9 illustrates a presentation that was generated following the specification and submission of a hypothesis to the databases. The hypothesis statement is displayed at the top of the screen. The set of graphs displayed beneath the hypothesis statement, depict the dependencies between parameters specified in the hypothesis and provide feedback to the user on whether or not there is any evidence to support their hypothesis. For the example hypothesis given above, two graphs are generated. One plots density against substrate width. The other plots efficiency against substrate width.



Resulting Graph

- The graph for the hypothesis is shown below.
- You can save the graph by *right-clicking* on the image and selecting *'Save As'*.
- Use the control box below to turn 'ON' the selection mode. Thereafter, you can highlight sections on the graph, where you would like to propose future experiments.
- Once you have made a selection, press the 'Propose Experiments' button and you'll be taken to a new screen where can provide values for any other paramter(s) needed for conducting future experiments.
- Also note that, if you zoom into SVG, the selection on graph will report incorrect values. So ensure that you are fully zoomed out, if you intend to make selections on graph. Reverting back to normal zoom ca the right-click menu on the SVG.

Selection Mode: ON

Plot of performance_testing.density against slip_batching.powder_content
performance_testing.density
5.5
5.45
5.4
5.35
5.3
5.25
5.2
5.15
5.1
5.05
5
59.5
59.6
59.7
59.8
59.9
60
60.1
60.2
60.3
60.4
slip_batching.powder_content
Entire Data
IF
IF-THEN trend

Propose Further Experiments

- Your graph selection is shown below
- You may refine the values of selection by manually editing them
- Once you are satisfied with the range of values, click 'Propose Experiments'
- Note that *performance_testing.density* is either an image attribute or an uncontrollable parameter. Since you cannot control it during experiments, its selection on the graph is not shown.

	From	To
slip_batching.powder_content	59.6243	59.845

Figure 10: Specification of New Experiments from Statistical Analysis of Data

In order to support more rigorous mathematical analysis of the data and in particular linear modeling of relationships between parameters, packages from the R statistical analysis software, were integrated into the data exploration and hypothesis testing system as web services. These enable curves to be fitted to the data and mathematical relationships to be determined between parameters. Figure 10 illustrates the results of fitting a curve to a plot of *density* versus *solid content*. GnuPlot [26] was used to plot the curves and to enable drawing and highlighting of 2D regions. 3D plots and regional specification are also possible. The blue region in Figure 10 highlights an area where more data is required. This range of solid content (the controllable parameter) is then used to initiate a number of new experiments and associated workflow instances

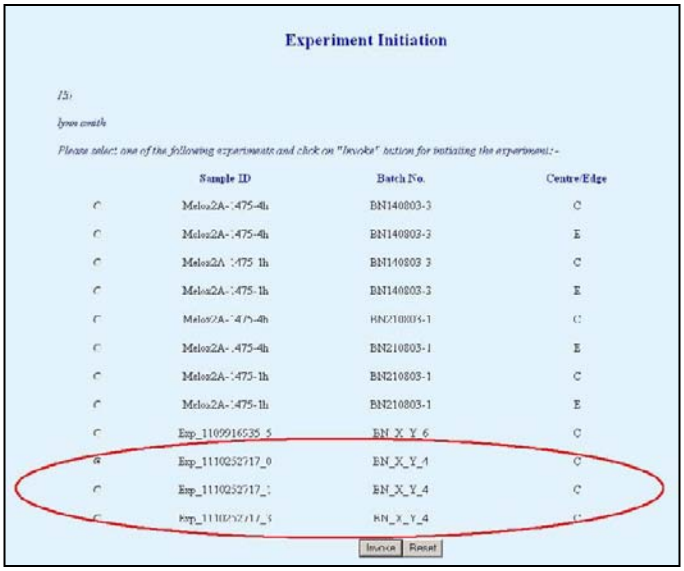
5. User Interface and System Functionality

When users login to the system they are authenticated via a login id and password. There are a range of different types of users who will require system access:

- Data analyst – this person uses the Data Exploration and Hypothesis Testing Interface and the R services to analyze the existing data in the MySQL databases and to specify where they would like more data to be captured. This generates a set of new experiments/workflow instances which are sent to the experiment initiator.
- Experiment initiator: the person with the authority to initiate an experiment/workflow by specifying the experimental parameters and the individuals responsible for each task. Figure 11 shows the interface that is displayed when this user logs on.
- Slip Batching, Tape Casting, Firing, Performance Testing and Microanalysis Users: these users are assigned specific tasks and are notified by email when they hnew or ave uncompleted tasks. The primary objective of each task is to capture data/metadata associated with a particular stage of the experiment
- Progress Tracker: this user has the authority to monitor the progress of the currently running workflows.

Users are notified by email when they have been allocated new tasks. When users login their list of open currently allocated tasks is displayed. When a user selects a particular experiment, the system displays all of the relevant parameters and their settings for that experiment. After the user has completed a task and saved the new data, it is validated against an underlying XML schema, stored in the relevant MySQL database and the system invokes the next task in the workflow.

For each workflow, there must be a business process written in BPEL4WS and an associated WSDL file [27]. Consider the workflow for the *electrolyte manufacture* process. This workflow consists of 5 sub-workflows: Slip_Batching.bpel, Tape_Casting.bpel, Firing.bpel, Microstructure.bpel and Performance_Test.bpel respectively. The BPEL4WS process itself is basically a flow-chart representation of an algorithm. When the sub-workflows are deployed to BPWS4J, they are treated as web services. A WSDL file is also required for the *electrolyte manufacture* business process since each deployed workflow is regarded as a web service. In this WSDL file, service link types are declared for defining the relationships between two (or potentially more) services. When two services interact with each other, the service link type is a declaration of how they interact – essentially what each party offers.



Experiment Initiation

Hi

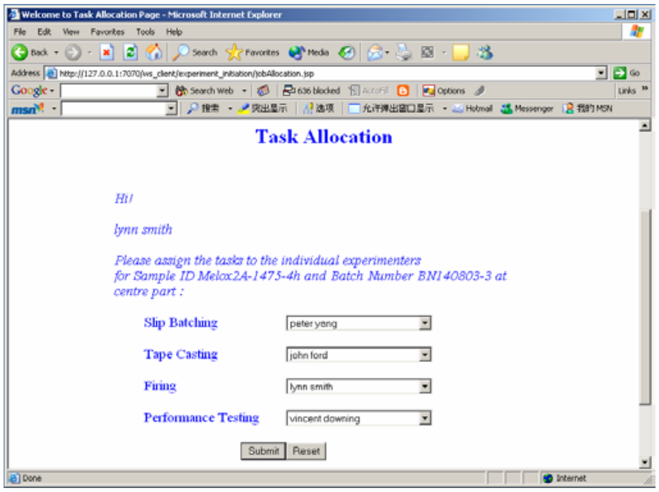
lynn smith

Please select one of the following experiments and click on "Invoke" button for initiating the experiment:-

	Sample ID	Batch No.	Centre/Edge
○	Melox2A-1475-4h	BN140803-3	C
○	Melox2A-1475-4h	BN140803-3	E
○	Melox2A-1475-1h	BN140803-3	C
○	Melox2A-1475-1h	BN140803-3	E
○	Melox2A-1475-4h	BN210803-1	C
○	Melox2A-1475-4h	BN210803-1	E
○	Melox2A-1475-1h	BN210803-1	C
○	Melox2A-1475-1h	BN210803-1	E
○	Exp_1109916535_5	BN_X_Y_6	C
●	Exp_1110252717_0	BN_X_Y_4	C
○	Exp_1110252717_1	BN_X_Y_4	C
○	Exp_1110252717_3	BN_X_Y_4	C

Figure 11: Set of New Experiments Initiated from Data Analysis Interface

Before the experiment initiator can actually invoke a new experiment, they must allocate the sub-tasks in the workflow. This will ensure that the relevant people are notified when they have been allocated new tasks and the appropriate controllable settings are specified according to the experimental design and workflow. Figure 12 illustrates the sub-task allocation interface.



Welcome to Task Allocation Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://127.0.0.1:7070/ws_client/experiment_initiation/jobAllocation.jsp
Go
Google Search Web 636 blocked AutoFill Options
Links
msn 搜索 突出显示 选项 允许弹出窗口显示 Hotmail Messenger 我的 MSN
Done
Internet

Task Allocation

Hi!

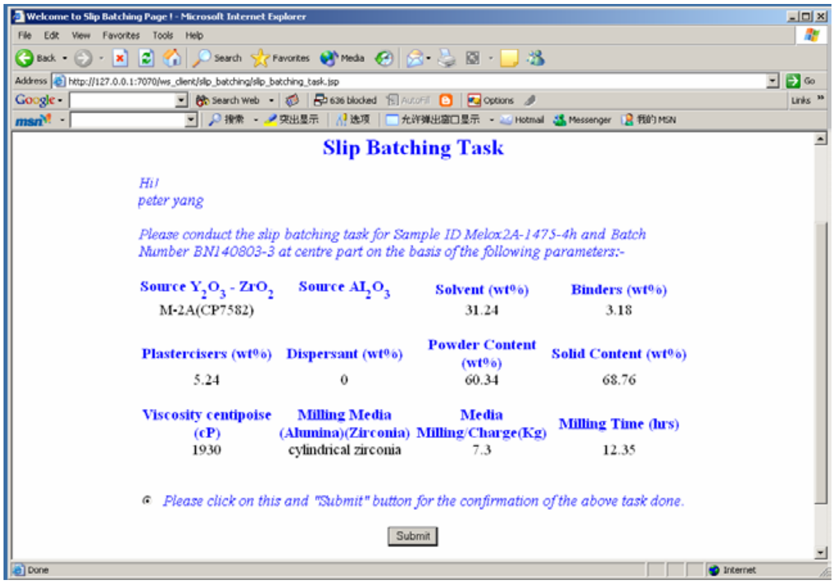
lynn smith

Please assign the tasks to the individual experimenters for Sample ID Melox2A-1475-4h and Batch Number BN140803-3 at centre part :

Slip Batching	peter yang
Tape Casting	john ford
Firing	lynn smith
Performance Testing	vincent downing

Figure 12: Sub-Task Allocation Interface for Experiment Initiator

Figure 13 illustrates the interface that Peter Yang, the person responsible for the Slip Batching task will see, when he logs onto the system. It clearly specifies all of the experimental settings that he must use. When Peter Yang has completed his task and input his results, they are validated, saved into the database and the next person (John Ford) is notified that he must complete the Tape Casting sub-task.


Welcome to Slip Batching Page ! - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://127.0.0.1:7070/ws_client/slip_batching/slip_batching_task.jsp
Go
Google Search Web 636 blocked AutoFill Options Links
msn 搜索 突出显示 选项 允许弹出窗口显示 Hotmail Messenger 我的 MSN
Done
Internet

Slip Batching Task

Hi!
peter yang

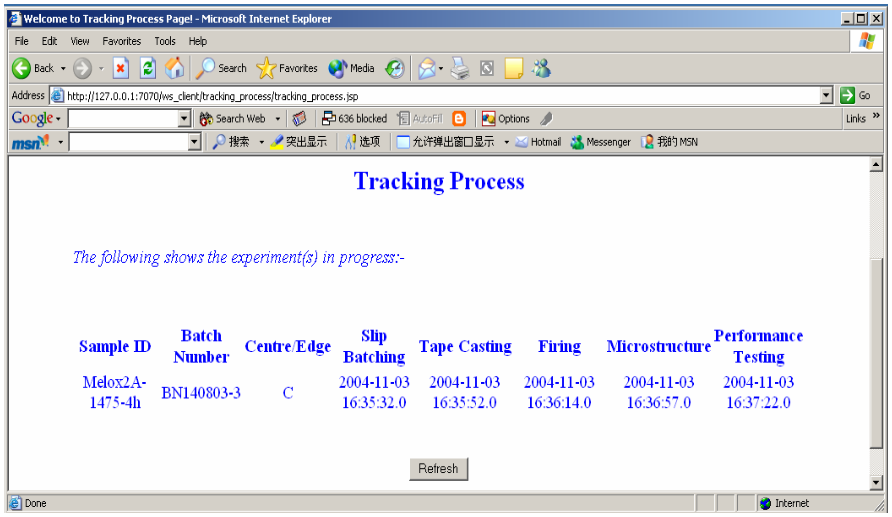
Please conduct the slip batching task for Sample ID Melox2A-1475-4h and Batch Number BN140803-3 at centre part on the basis of the following parameters:-

Source Y_2O_3 - ZrO_2	**Source AI_2O_3**	**Solvent (wt%)**	**Binders (wt%)**
M-2A(CP7582)		31.24	3.18
Plastercisers (wt%)	**Dispersant (wt%)**	**Powder Content (wt%)**	**Solid Content (wt%)**
5.24	0	60.34	68.76
Viscosity centipoise (cP)	**Milling Media (Alumina)(Zirconia)**	**Media Milling/Charge(Kg)**	**Milling Time (hrs)**
1930	cylindrical zirconia	7.3	12.35

● *Please click on this and "Submit" button for the confirmation of the above task done.*

Submit

Figure 13: User Interface for Person Responsible for Slip Batching Task


Welcome to Tracking Process Page! - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://127.0.0.1:7070/ws_client/tracking_process/tracking_process.jsp
Go
Google Search Web 636 blocked AutoFill Options Links
msn 搜索 突出显示 选项 允许弹出窗口显示 Hotmail Messenger 我的 MSN
Done
Internet

Tracking Process

The following shows the experiment(s) in progress:-

Sample ID	Batch Number	Centre/Edge	Slip Batching	Tape Casting	Firing	Microstructure	Performance Testing
Melox2A-1475-4h	BN140803-3	C	2004-11-03 16:35:32.0	2004-11-03 16:35:52.0	2004-11-03 16:36:14.0	2004-11-03 16:36:57.0	2004-11-03 16:37:22.0

Refresh

Figure 14: Workflow Monitoring User Interface

At any stage, users with the appropriate authority are able to see the current state of workflow instances (i.e., currently running experiments) through an experiment monitoring interface.

6 System Evaluation

User testing of the system has been carried out by fuel cell scientists from The University of Queensland's Centre for Microscopy and Microanalysis. Feedback from the users to date has indicated the following:

- The user interface design and incorporation of domain-specific ontologies allowed users with little knowledge of the domain, to quickly and easily explore the data and gain an understanding of the knowledge space;
- Different users carry out research activities differently. Being able to customize or personalize the mode, scope and focus of the assimilated data presentations and the hypothesis refinement process was extremely beneficial for individual productivity;
- The different presentation modes enabled faster processing and interpretation of large data sets and images by the fuel cell scientists than was possible manually and expedited the hypothesis generation and refinement process;
- Being able to record, browse and retrieve past investigations and hypotheses, reduced duplication and enabled existing hypotheses to be refined or new hypotheses to be developed based on past work. It also provides a way of capturing and sharing tacit domain expert knowledge, explicitly, in the form of rules;
- Statistical analysis techniques are only applicable to quantitative data. A major advantage of The FUSION system's approach is that by integrating statistical analysis with visualization tools and workflow generation, it applicable across a range of data and media types including images.
- The use of semantic web technologies such as ontologies, annotations and inferencing rules, provide a consistent, machine-processable way for describing, capturing, reusing and building on the domain knowledge. It also enables better collaboration between distributed research laboratories and industry through improved sharing of knowledge and data
- The use of Web Services offers a more flexible dynamic approach that enables maximum reuse of existing software modules and enables interoperability between applications written in different languages e.g., BPWS4J in Java and MATLAB in C. New emerging technologies such as Web Services Resource Framework (WSRF) [3], Web Services Choreography Interface (WSCI) [28] and OWL-S (Ontology Language for Web Services) [29] are expected to enable even more dynamic and intelligent composition of web services to automatically generate eScience workflows in the future.

7 Conclusions and Future Work

In this paper we describe how a particular scientific experimental workflow (the *electrolyte manufacture* process) can be represented as a business process, defined using BPEL4WS and executed using BPWS4J. We then describe how this is used to capture the data and precise provenance metadata associated with an experimental process, validate it and store it within distributed databases for further analysis, sharing and correlation.

We then describe a search interface that enables scientists to interact with a knowledge base through a hypothesis-driven approach that combines data exploration, integration, search and inferencing. Furthermore, by integrating statistical data analysis methods into the interface and applying them to formulated hypotheses, users are able to fit more precise mathematical models to relationships between parameters. In addition, they are able to determine mathematically where further data is required and generate new instances of experimental workflows – thus completing the knowledge cycle.

We believe that such systems represent the next generation of scientific knowledge management and that they are will be increasingly in demand and applied across many domains including engineering, homeland security, social sciences and health, to solve complex problems and provide decision support tools based on the analysis and assimilation of large-scale, mixed-media, multi-dimensional information and data sets. Plans for future work include:

- Further testing and refinement of the system, particularly within a real-world industrial environment. We plan to deploy it within a fuel-cell manufacturing company to facilitate the exchange of knowledge between university research and industry organizations in this domain;
- Investigating how the empirical modeling approach described here can be combined with the physical modeling approach to generate a more accurate predictive model for simulating fuel cell behaviour.
- Testing the portability, flexibility and scalability of the system by applying it to other domains, such as environmental modeling and bioinformatics.

Acknowledgements

The work described in this paper has been funded by the Cooperative Research Centre for Enterprise Distributed Systems Technology (DSTC) (through the Australian Government's CRC Programme) and through a University of Queensland Development Grant. Thanks to John Drennan and the University of Queensland's Centre for Microscopy and Microanalysis for providing the fuel cell images and data and for their valuable user feedback.

References

[1] Andres, T.F. Cubera, et al (2003). *Specification: Business Process Execution Language for Web Service Version 1.1*, BEA, IBM, Microsoft, SAP AG and Siebel System.
[2] *Open Grid Services Infrastructure (OGSI) Version 1.0* http://www.gridforum.org/documents/GFD.15.pdf
[3] *Web Services Resource Framework* *http://www.globus.org/wsrf/*
[4] *FUSION* Project http://metadata.net/sunago/fusion.html
[5] *MATLAB*, http://www.mathworks.com/ [6 October 2004]
[6] *The R Project for Statistical Computing* http://www.r-project.org/ [26 October 2004]
[7] IBM-alphaWorks. *BPWS4J* http://www.alphaworks.ibm.com/tech/bpws4j [6 October 2004]

[8] Singh and Vuok, *Scientific Workflows: Scientific Computing Meets Transactional Workflows,* http://www.csc.ncsu.edu/faculty/mpsingh/papers/databases/workflows/sciworkflows.html

[9]M.Greenwood, *Comparing Workflow in eScience and eBusiness,* May 2004 http://twiki.mygrid.org.uk/twiki/bin/view/Mygrid/WorkFlowDifferences

[10] The official website of ^{my}*Grid project* http://www.mygrid.org.uk/ [6 October 2004]

[11] Addis, M, J. Ferris et al (2003) *Experiences with eScience workflow specification and enactment in bioinformatics*, In Proceedings of the UK e-Science All Hands Meeting 2003, Nottingham UK. pp. 459-466.

[12] The official website of *Taverna* http://taverna.sourceforge.net/ [26 October 2004]

[13] *Freefluo* http://freefluo.sourceforge.net/ [26 October 2004]

[14] *Kepler* http://kepler.ecoinformatics.org/ [8 October 2004]

[15] *Ptolemy II*, http://ptolemy.eecs.berkeley.edu/ptolemyII/ [08 October 2004]

[16] Altintas, I, C. Berkley et al *Kepler: An Extensible System for Design and Execution of Scientific Workflows* 16th Intl. Conference on Scientific and Statistical Database management (SSDBM), Santorini Island, Greece, June 2004

[17] Aalst, W., L. Aldred et al *Design and Implementation of the YAWL system* In Proceedings of the 16th International Conference on Advanced Information Systems Engineering (CAiSE 04), Riga, Latvia, June 2004

[18] Aalst, W, A. Hofstede et al *Workflow Patterns* BETA Working Paper Series, WP 47, Eindhoven University of Technology, Eindhoven, 2000.

[19] *Apache Tomcat* http://jakarta.apache.org/tomcat/ [6 October 2004]

[20] *Apache Axis*, http://ws.apache.org/axis/ [7 October 2004]

[21] *MySQL* http://www.mysql.com/ [6 October 2004]

[21] *JavaMail API*, Sun Microsystems, http://java.sun.com/products/javamail/ [6 October 2004

[22] Microsoft, HTML+TIME 2.0 Reference, (http://msdn.microsoft.com/workshop/author/behaviors/reference/time2_entry.asp).

[23] W3C, Scalable Vector Graphics (SVG) 1.1 Specification, W3C Recommendation, 14 January 2003, Fujisawa Jun and Dean Jackson Edited by Jon Ferraiolo, (http://www.w3.org/TR/SVG/).

[24] W3C, OWL Web Ontology Language Reference, W3C Candidate Recommendation, 18 Aug 2003, Edited by Mike Dean and Guus Schreiber, (http://www.w3.org/TR/owl-ref/).

[25] Hunter, J., Drennan, J., and Little, S., *Realizing the Hydrogen Economy through Semantic Web Technologies.* IEEE Intelligent Systems Journal - Special Issue on eScience. 2004.

[26] GnuPlot (http://www.gnuplot.info/)

[27] W3C. *Web Services Description Language (WSDL)* W3C Note, http://www.w3.org/TR/wsdl [6 October 2004]

[28] W3C. *Web Service Choreography Interface 1.0* http://www.w3.org/TR/wsci/ [27 October 2004]

[29] *Ontology Web Language for Services (OWL-S) Version 1.0* http://xml.coverpages.org/ni2004-01-08-a.html [27 October 2004