ESRC e-Infrastructure: Workflow Requirements and Evaluation

Rob Allan
Computational Science and Engineering Department,
Daresbury Laboratory, Daresbury, Warrington WA4 4AD

Dexter Canoy
NIBHI, School of Community based Medicine,
University of Manchester, University Place,
Oxford Road, Manchester M13 9PL

Andy Turner
CCG, School of Geography, University of Leeds
Woodhouse Lane, Leeds LS2 9JT

Contact e-Mail: `r.j.allan@dl.ac.uk`, `dexter.canoy@manchester.ac.uk`,
`a.g.d.turner@leeds.ac.uk`

Abstract

This report is a deliverable of the ESRC e-Infrastructure Project WP3.4.1.

The target audience is e-social science developers, advanced users, deliverable D3.4.2, the JISC eReSS and NeISS projects and the e-Framework initiative (who will be interested in workflow use cases).

Scientific workflow has become attractive in recent years as science becomes increasingly reliant on the analysis of massive data sets and the use of distributed resources. The workflow programming paradigm is seen as a means of managing the complexity in defining the analysis, executing the necessary computations on distributed resources, collecting information about the analysis results and providing means to record and reproduce the scientific analysis.

This report presents a very brief and incomplete overview of the current state of the art in the field. We nevertheless hope it will be useful to illustrate how workflow tools are used in e-Social Science. It references research from many of leading computer scientists in the workflow area and provides real world examples from domain scientists actively involved in e-Social Science. The computer science topics addressed provide a broad overview of active research focusing on the areas of workflow representations and process models, component and ervice based workflows, standardization efforts, workflow frameworks and tools, and problem solving environments and portals.

The topics covered represent a broad range of scientific workflow and should be of interest to a wide range of computer science researchers, domain scientists interested in applying workflow technologies in their work, and engineers wanting to develop workflow systems and tools.

Neither the STFC e-Science Centre nor its collaborators accept any responsibility for loss or damage arising from the use of information contained in any of their reports or in any communication about their tests or investigations.

Contents

1 Introduction

This report is a deliverable of the ESRC e-Infrastructure Project WP3.4.1.

The target audience comprises of e-Social Science developers, advanced users, deliverable D3.4.2 and the JISC e-Framework initiative (who will be interested in workflow use cases).

The aims and objectives of this work package are:

- To determine how workflows are used within e-Social Science and develop use cases;
- To investigate the use of workflows within e-Social Science;
- To deploy sample e-social science workflows;
- To compare workflow tools for their suitability to e-Social Science;
- To determine user requirements of workflows.

Much of the work of social scientists and policy makers is consumed by accessing, collating and analysing data. This is particularly true in the domains of urban planning and economic modelling. Unfortunately, the tools which can facilitate this process are not well understood and much of the integration is still done manually by *ad hoc* methods. Moreover, raw data are of limited utility. Usually these data sets are the input to models of more complex phenomena that produce additional data of interest. For example, in modelling traffic flow, we can derive lorry traffic along specific motorway links within a metropolitan area based on quite far removed raw (source) data such as employment, imports into and exports out of the region, etc. by using a complex workflow of operations.

There is an increasing need for workflow capabilities across both humanities and science sectors. In using software and online resources, researchers in these sectors typically carry out tasks involving the design and execution of a series of steps, or workflow. A researcher begins by identifying and accessing initial data sets and proceeds through additional steps using software tools such as Web services, modelling and simulation programs, image processing programs, visualization software, etc. This is encapsulated in the research life cycle. Each of these steps progressively transforms the initial data, and researchers need to keep track of what was done and why. Adding to the complexity, researchers are often attempting to run quantitative and repeatable analyses and models in more than one software and hardware environment.

A typical experimental research activity [28] involves the following steps: observation, hypothesis, prediction (under specified constraints), experiment, analysis and write-up. This was extended in a JISC study in 2007 [7] to include the steps as shown in Figure 1.

As a sample of a computer based scientific procedure which could be expressed as a workflow, we show in Figure 2 how the RMCS suite of software [27] developed in the NERC funded eMinerals e-Science project is now being used for parametric studies in the design of new materials in CCP9, the Collaborative Computational Project number 9 (Electronic Structure of Materials).

Whilst workflow technologies provide support for researchers to define an "experiment", there is no support for capturing the constraints associated with it, therefore making it difficult to situate the

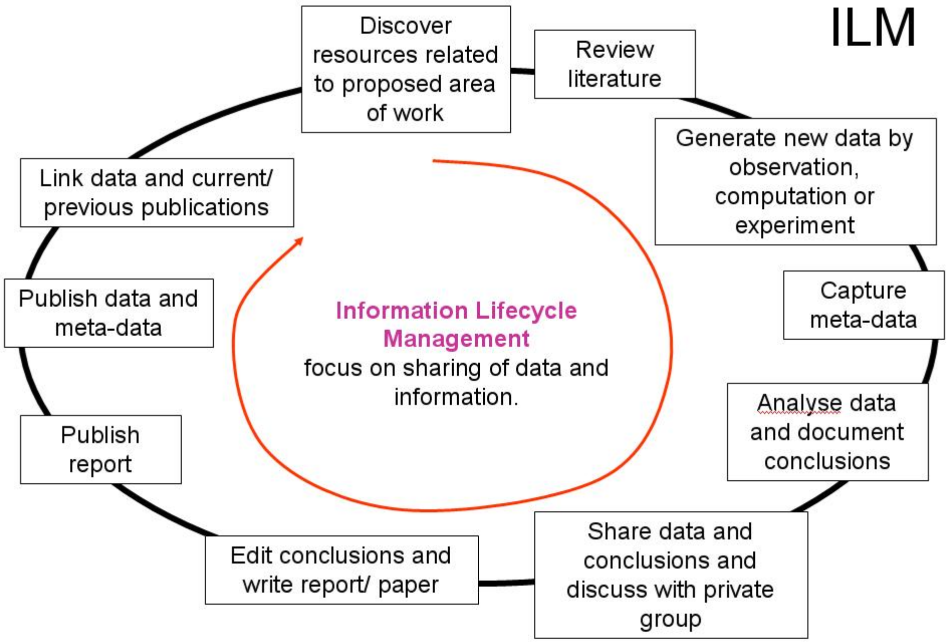
ILM
Discover resources related to proposed area of work
Review literature
Generate new data by observation, computation or experiment
Capture meta-data
Analyse data and document conclusions
Share data and conclusions and discuss with private group
Edit conclusions and write report/ paper
Publish report
Publish data and meta-data
Link data and current/ previous publications
Information Lifecycle Management
focus on sharing of data and information.

Figure 1: e-Research Data and Information Lifecycle Management

experiment in context. Pignotti *et al.* [24] argued that in order to characterise scientific analysis we need to go beyond low level service composition and execution by capturing a higher level description of the experimental process. The aim here is to make the prerequisites and goals of the experiment, which we describe as the “scientist’s intent”, transparently.

RMCS does not aim to provide a general purpose workflow tool and is script based rather than graphical. It uses the underlying Condor DAGMan (Directed Acyclic Graph Manacer) and Condor-G to invoke Grid services. Six true workflow tools will be compared in this report to determine how they meet user requirements. These are briefly summarised as follows [22]:

WS-BPEL: Web Services Business Process Execution Language, a standard widely used in e-Business systems, see `http://en.wikipedia.org/wiki/BPEL`.

Kepler: The Kepler project’s overall goal is to produce an open source scientific workflow system that allows scientists to design research workflows and execute them efficiently using emerging Grid based approaches to distributed computation. Kepler is based on the Ptolemy II system for heterogeneous, concurrent modelling and design, see `http://kepler-project.org`.

Kuali: The Kuali Foundation supports the development of free and open source administrative software used to manage workflows in academic administrative processes. It was originally developed at Indiana University. Kuali’s modular design includes a base system of Chart of Accounts, General Ledger, Transactions, Reporting and Workflow. Additional modules that can be implemented as needs are identified include: Accounts Receivable, Budgeting, Capital Assets Management, Endowment, Enhanced Decision Support and Reporting, Labour Distribution, Purchasing

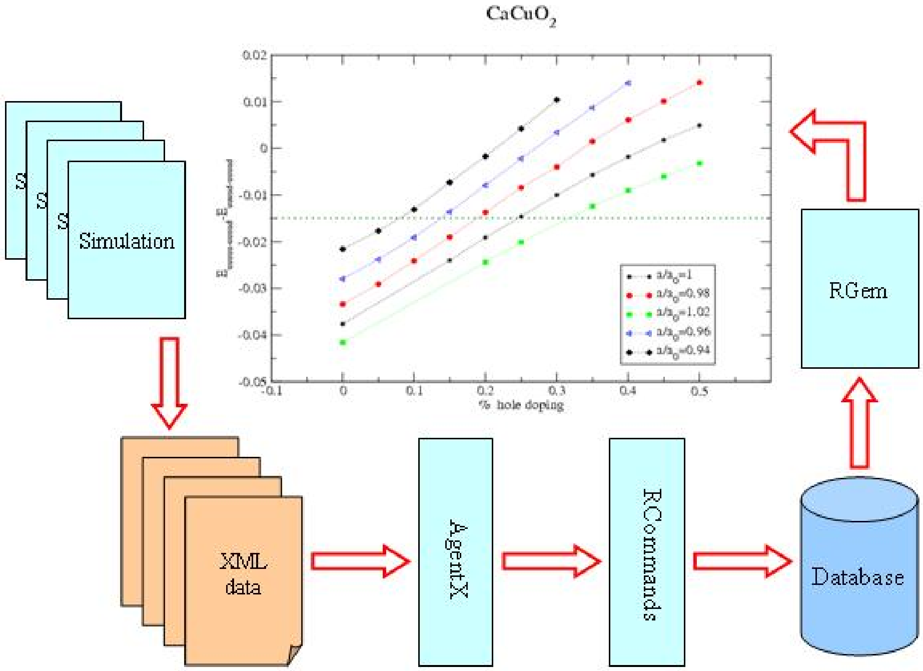

Figure 2: Grid-based Parametric Studies for e-Minerals and CCP9 Projects.

and Accounts Payable, Pre- and Post-Award Research Administration. See `http://kuali.org`.

Pegasus: Planning for Execution in Grids, is a workflow mapping engine developed and used as part of several NSF projects in the USA. Pegasus bridges the scientific domain and the execution environment by automatically mapping the high level workflow descriptions onto distributed, see `http://pegasus.isi.edu`.

Taverna: The Taverna Workbench from the EPSRC funded myGrid e-Science project allows users to construct complex analysis workflows from components located on both remote and local machines, run these workflows on their own data and visualise the results, see `http://taverna.sourceforge.net` and `http://www.mygrid.org.uk`.

Triana: An open source problem solving environment developed at Cardiff University that combines an intuitive visual interface with powerful data analysis tools. Triana is already used by scientists for a range of tasks, such as signal, text and image processing. It includes a large library of pre-written analysis tools and the ability for users to easily integrate their own tools, see `http://www.trianacode.org`.

Finally we note that workflow in this sense only really works if the process can be described in a service oriented way. We will therefore assume that each component can be invoked as a Web service, with its business description and interface listed in a WSDL registry, possibly with some additional semantic information meaningful to the domain scientist. There are many access and security issues associated with such an SOA based workflow. These issues will not be addressed here, but are partly the subject of the new JISC funded NeISS project, National e-Infrastructure for Social Simulation, see `http://www.neiss.org.uk`.

2 Methodologies used for Requirements Gathering

We are not proposing to do a survey, but have sought input from other participants in the e-Infrastructure project and elsewhere.

Current discussions have been with: Andy Turner, Paul Townend, Junaid Arshad, Xiabo Yang, Wei Jie, Pascal Ekin.

We have also sought input from PolicyGrid, mainly in the form of analysis of their published papers.

We have discussed this WP with Asif Akram, an expert on workflow who was the principal RA on WOSE, the EPSRC funded project Workflow Optimisation for Services in e-Science. For more information on this see `http://www.grids.ac.uk/WOSE`.

Finally we have ongoing discussions with developers on the myExperiment project who are also involved in NeISS, see `http://www.myexperiment.org`.

3 E-Social Science Workflow Use Cases

Four use cases arising from the requirements gathering focus our initial discussions.

3.1 MoSeS

MoSeS is an NCeSS node focussing on development of a national demographic model and simulation of the UK population specified at the level of individuals and households. MoSeS helps town planners to forecast trends in healthcare, business and transport in policy making by predicting demographic changes looking forward up to 20 years.

To make predictions, computationally intensive agent based simulation models are run on the NGS using a number of distributed data sources. MoSeS links into services such as the Census for current and historical demographic information and the EDINA geo-linking service for mapping data on different regions such as electoral districts known as wards.

MoSeS offers users a number of different ways to interact with its simulations and scenarios. A series of JSR 168 portlets have been developed for the MoSeS portal which currently uses the GridSphere framework. The same portlets will work in the e-Infrastructure Sakai VRE. From this interface, PDF reports of simulation results can be generated via a simple workflow. These reports contain maps, tables, comparisons, etc. based on user defined criteria. It is also possible to visualise simulation results using mashups with the Google Maps service or streamed results into Google Earth.

We are currently using the use case of creating a report to compare planning scenarios for future health care policy. This is based on the portlet approach currently implemented for MoSeS. The workflow was discussed at the meeting with the MoSeS project in Manchester 8/1/2008. The following was noted in relation to this WP.

- Paul Townend demonstrated the sequence of portlets used to query data, generate graphs and tables which could be used as input into policy reports, e.g. for future healthcare requirements;
- if the underlying logic is converted to Web services, this could form the basis of a workflow – Paul agreed that this was a good approach;
- there are other RESTful services available, e.g. the GLS service invoking EDINA services which is now installed in Sakai for the project;
- running through the sequence by hand in the portlet, we could capture the workflow and then re-enact it with other data or forecasts to compare scenarios and automate report production;
- Pascal Ekin noted that a GIS compatible visualisation tool might be nice;
- Pascal also suggested writing a boundary converter, e.g. to convert data from wards to healthcare regions;

3.2 Obesity e-Lab

Obesity is now a major public health problem. The rising number of obese children and adults means that there are far reaching consequences not just to the individual but also to the society. The obesity epidemic is poorly understood but societal factors are known to influence behaviours and lifestyles that may pre-dispose the individual to gain excess weight. A need for obesity research to be collaborative across disciplines, such as social science, public health and epidemiology, has been recognised. However, unless there are shareable research processes that facilitate insights across disciplines, inter-disciplinary research, such as in obesity, is unlikely to succeed. The Obesity e-Lab aims to enable "obesity researchers" from various disciplines to share data, information and analytical tools. Underpinning this project is the development of an e-Lab architecture to allow a secure environment for producing, finding, sharing and implementing a "research object", which is a package of digital resources required to reproduce and communicate a research finding. The Obesity e-Lab will be designed to reduce the barrier of accessing and using publicly available databases, and facilitate the use of these data sources for obesity research. We aim to develop a Web based application to facilitate obesity research using publicly available data sources, such as the Health Survey for England, by implementing the following.

1. allow access to registered users to relevant databases securely;
2. identify variables (and all related contextual information) across survey years;
3. create metadata of selected variables, capture data management processes such as data cleaning, derivation of new sets of variables, etc. and record analytical or statistical scripts;

4. provide visualisation methods at various stages of the research process to aid interpretation.

These processes will be captured and digitised as a research object which can be used to share, reproduce or replicate research process or findings and even re-use or re-purpose aspects of the research process to answer other research questions. The Obesity e-Lab could serve as a place to find and explore relevant data for obesity research and a place to collaborate through sharing methodologies and publishing results for others to use. Although this project is aimed to facilitate research in obesity related topics, the underlying architecture of bringing together datasets, investigators and methods, and their packaging into research objects for sharing and provenance purposes, are generic and potentially useful for other social research topics.

We note that Obesity e-Lab is working with the myExperiment project to use Taverna.

3.3 PolicyGrid

A third use case is taken from PolicyGrid [24]. Computer scientists and social scientists at the University of Aberdeen, the Macaulay Institute (Aberdeen) in the PolicyGrid project are investigating the use of semantic workflow tools to facilitate the design, execution, analysis and interpretation of simulation experiments and exploratory studies, while generating appropriate metadata automatically.

Recent activities in the field of social simulation [25] indicate the need to improve the scientific rigour of agent based modelling. One of the important aspects of science is that work should be repeatable and verifiable. Results gathered from possibly hundreds of thousands of simulation runs cannot yet be reproduced conveniently in a journal publication. Equally, the source code of the simulation model and full details of the model parameters used are also not journal publication material. PolicyGrid have identified activities that are relevant to such situations.

- being able to access the results, to check that the authors' claims based on those results are justifiable;
- being able to re-run the experiments to check that they produce broadly the same results;
- being able to manipulate the simulation model parameters and re-run the experiments to check that there is no undue sensitivity of the results to certain parameter settings;
- being able to understand the conditions in which the experiment was carried out.

Pignotti et al. [24] used Kepler to access various services in a virus simulation model as part of a scenario for studying the transmission and perpetuation of a virus in the human population. The workflow tool enables the simulation to be repeated under the assumption of various conditions.

3.4 NaCTeM; National Centre for Text Mining

The Liverpool group (Watry et al.) have investigated the use of workflows to underpin some activities of NaCTeM. This was the subject of a JISC funded VRE-1 project `http://www.jisc.ac.uk/whatwedo/programmes/programme_vre/cheshire3_vre.aspx`.

The project sought to develop and implement the Kepler/ Ptolemy scientific workflow system as an interface to the Cheshire-3 digital library framework. The aim was to enable researchers in both the humanities and scientific disciplines to use the Kepler/ Cheshire software to conduct analyses and perform distributed processing in several different software and hardware environments; and to coordinate the export and import of data from one environment to another. They intended to use the Kepler/ Cheshire interface to provide researchers with capabilities ranging from discovering information to publishing results, thus comprising a Virtual Research Environment. In particular, they intended to work with AHDS, the Arts and Humanities Data Service, to develop a number of transactional services for the humanities.

The overall aim of the project was to implement established, automated workflow technologies into the Cheshire-3 digital library framework. This would provide researchers with an easy to use yet powerful system for executing workflows and would enable users of the system to generate, more easily, publishable results from relevant text data.

The outputs of the project could form a major contribution to the teaching and research environments across a spectrum of projects and services, particularly multi-disciplinary projects involving resources in the humanities. For example, it could allow users of AHDS and other services to automate complex workflows without having to become expert programmers.

This capability should relieve researchers of repetitive tasks so that they can focus on their particular area of expertise. It will also give researchers increased capabilities to communicate and work together – searching for, integrating and sharing data and workflows in large scale collaborative environments. In effect, the project will add transactional capabilities to services which are at present focused on content. We note that myExperiment also has similar aims.

The specific objectives were as follows.

- devise an interface for a workflow creation and execution process so that users may design, execute, monitor, and communicate analytical procedures repeatedly with minimal effort;
- implement this as part of the Cheshire-3 digital library framework;
- incorporate this integration into data Grid systems, through support of the Storage Resource Broker and Grid workflow patterns;
- in doing so, address issues of data and process provenance, user interaction, reporting and logging;
- test the implementation on large, complex, and heterogeneous data sets particularly from AHDS;
- Evaluate the implementation with improvements introduced from user feedback.

There are three reasons why Kepler and Ptolemy was selected for this work.

1. it is the only available system which allows one to plug in different execution models into workflows;
2. it is a mature system which is already widely used and supported in the e-Science and cyber infrastructure communities;

3. it leverages related joint development work that the Liverpool group are undertaking with the San Diego Supercomputer Center, e.g. work on SRB.

In terms of Kepler/ Cheshire-3 integration, the technical steps were described as follows.

- Configure a Kepler Web service actor to use SRW (Search and Retrieve Web service). We first need to investigate whether the Ptolemy/ Kepler compiler will be able to compile the WSDL (Web Service Definition Language) for SRW, which has proven to be more complex than conventional transactional services. If this does not work with the Ptolemy/ Kepler compiler, then we will need to build the object definitions by hand or treat it as document literal and use regular XML processing on the back end to parse the response. This will let us use Kepler to interact with Cheshire as a "black box" – eg all you can do are the operations allowed by SRW (search and scan, primarily). If possible the actor should self configure from the Explain response, but that is not prioritised.
- Write a series of Kepler actors to interact directly with Cheshire-3 objects – for example a PreParser actor, a Parser actor, a Transformer actor and so forth. This will be more complex than the previous Web service based interaction and will require linking the Java based Kepler with the Python based Cheshire-3. The expected implementation will rely on TCP/ IP sockets so as to be distributable in the Grid environment. Transporting objects around the network will require serialisation, potentially using Python's fast "pickle" algorithm in a similar method to the distributed processing module for PVM or MPI. The requirement to maintain Session objects between calls however may mean that it becomes easier for Kepler to interact with a Python daemon rather than with the Cheshire-3 objects directly.
- Write a Cheshire-3 to Kepler handler to be instantiated primarily as PreParser, Transformer and possibly a Normaliser. This will let us call Kepler workflows from Cheshire and import the results back in to the local environment. For this we need some link between a constantly running Java Kepler execution server and the Cheshire-3 objects that need to call them.

The project can support an environment which will allow researchers across a range of disciplines to analyse information discovered using the Cheshire-3 digital library system. It will make available to humanities researchers a number of important tools originating from the e-Science community and for the scientific researchers it will integrate these tools with a digital library framework. Strategically, it will introduce a range of transactional services to the humanities sector, including visualisation technologies, data mining systems and use of statistical programmes. Finally, it will investigate the implementation of distributed Grid services to enable researchers to use computational resources on the Internet in a distributed workflow.

4 User Requirements for Workflow Tools

A number of user requirements can be identified from the above cases.

- workflow system should remove the burden of programming for non-experts allowing them to focus on their research;

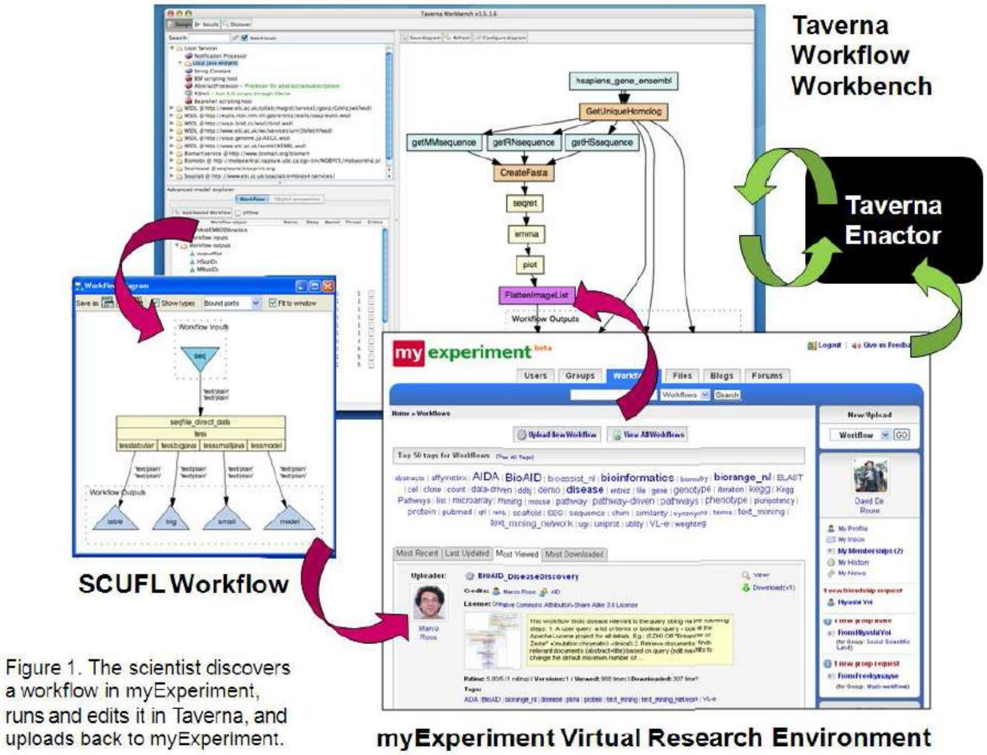
Taverna Workflow Workbench
Taverna Enactor
hsapiens_gene_ensembl
GetUniqueHomolog
getMMsequence
getRNsequence
getHSsequence
CreateFasta
seqret
emma
plot
FlattenImageList
Workflow Outputs
SCUFL Workflow
myExperiment Virtual Research Environment
Figure 1. The scientist discovers a workflow in myExperiment, runs and edits it in Taverna, and uploads back to myExperiment.

Figure 3: Research Lifecycle in myExperiment

- a visualisation tool or GUI is required;
- the workflow system should permit modification to repeat the original experiment or re-run the process on similar data;
- the underlying services in a workflow need to be provided with a standard interface, e.g. WS-I. They need to be published so that they can be located and composed using a graphical workflow editor;
- there are many issues of access and secrity which need to be identified and addressed if workflows are to be shared;
- the workflow plus its inputs and outputs should be capable of encapsulation along with semantic and packaging information as a "research object" which can be published;
- more than one workflow tool is in use and these can be installed on the user's desktop or on a server. Ideally workflows should be portable or a "broker" should be available;
- workflows could be published, e.g. as in myExperiment, and available to be enacted on the Internet, e.g. alongside other tools and information sources in a portal.

The benefits of sharing workflows and scripts motivated the creation of the myExperiment social Web site as funded under the JISC VRE-2 Programme. Using myExperiment, researchers can share their “experiments” whatever their workflow system. Figure 3 depicts the cycle of workflow discovery in myExperiment and editing in Taverna as an example. MyExperiment could be described as “Facebook for scientists”, but is different to sites for sharing pictures, slides, etc, because it focuses on the specific requirements of researchers, such as the need to describe the attribution of work, control visibility and sharing in groups, handle licensing and work with distributed collections of data.

5 Comparison of Workflow Tools

Many of the concepts underlying today’s e-Science workflow technologies originated from business workflows. These typically describe the automation of a business process, usually related to a flow of documents. They were introduced to manage manufacturing processes in the early part of the 20th Century and into sofware in the 1980s and are now being considered for automation of research processes which have to be managed, recorded or repeated. Scientific workflow is about the composition of structured activities, e.g. database queries, simulations, data analysis activities, etc. that arise in scientific problem solving. However, the underlying representation of the workflow remains the same (data and control flow). In this section we restrict our discussion to BPEL, Kepler, Pegasus and Taverna.

5.1 BPEL

The BPEL language [29] was originally designed for business, but has been adapted for scientific workflow use. BPEL4WS is an extension of BPEL and provides a language for the formal specification of processes by extending the Web services interaction model to enable support for business transactions. BPEL4WS specifies how to connect multiple Web services to provide a new Web service. The workflow is executed in terms of blocks of sequential service invocations. The main limitation of BPEL is that it does not support the use of semantic metadata to describe the workflow components and their interaction but instead relies entirely on Web services described by WSDL (Web Service Description Language). This type of language in not the best fit for our solution as we need rich metadata support for the workflow to describe not only service related information (e.g. platform, inputs and outputs) but also high level concepts (e.g. virus, population and model). The proposed use of research objects including metadata could however rectify this omission.

5.2 Kepler

The Kepler initiative has developed a generic tool and environment that builds on existing technologies and will work in a wide range of applications to capture, automate, and manage researchers’ actions as they carry out workflows. The Cheshire VRE project is building on this tool by providing a range of digital library capabilities, enabling researchers to discover information from JISC services and, once found, re-use this information in various ways.

Kepler is actually a set of workflow steps designed to operate in the Ptolemy data flow system and includes aspects such as Web service interaction.

MoML [31] used in Ptolemy is a language for building models as clustered graphs of entities with inputs and outputs. Clustering in MoML allows reuse of subgraphs as discrete entities in a larger graph, but MoML does not provide any semantics for the connections between entities in the graph. Like Taverna with XScufl, Kepler [32] is a workflow tool based on the MoML language and Ptolemy-II system for heterogeneous, concurrent modeling and design.

Kepler provides a GUI which allows "drag and drop" creation and execution of workflows for distributed applications using the abstract "actor" of Ptolemy-II as a wrapper. Web and Grid services, Globus Grid jobs and GridFTP can be used as components in an application. There are several libraries of actors for different purposes and custom actors can be added by the user. The creation of composite actors consisting of a workflow of other actors is also possible using the graph clustering facility in MoML. Kepler extends the MoML language by using "directors" which define execution models and monitor the execution of the workflow. The use of clustering in the MoML specification allows different execution strategies to be mixed in one workflow. Kepler also supports the use of ontologies to describe actors' inputs and outputs, enabling it to support automatic discovery of services and facilitate the composition of workflows.

Like other workflow tools, Kepler does not allow the use of metadata at runtime. However, the director component and the integration of ontologies with workflow activities provide an ideal interface within which the framework can operate.

Kepler has been used in the contest of a social science application by Pignotti et al. [24]. Kepler has also been used along with the Cheshire digital library software and SRB in the JISC VRE-1 project as described above.

5.3 Pegasus

Gil *et al.* [30] present some interesting work on generating and validating large workflows by reasoning on the semantic representation of workflow. Their approach relies on semantic descriptions of workflows templates and workflow instances. This description includes requirements, constraints and data products which are represented in ontologies. This information is used to support the validation of the workflow but also to incrementally generate workflow instances. Pignotti *et al.* not that in their research they are not focusing on assisted workflow composition, but do share the same interest in the benefit of enhanced semantics in workflow representation. While both these approaches rely on logical statements that apply to workflow metadata, PolicyGrid are taking a more user centred approach by capturing higher level methodological information related to scientist's intent, e.g. valid simulation result, epidemic virus, etc.

5.4 Taverna

Taverna [23] is a tool developed by the EPSRC funded myGrid e-Science project project to support "in silico" experimentation in biology, which interacts with arbitrary services that can be wrapped around Web services. It provides an editor tool for the creation of workflows and the facility to locate services from a service directory with an ontology-driven search facility. The semantic support in Taverna allows the description of workflow activities but is limited to facilitating the discovery of suitable services during the design of a workflow. Our scientist's intent framework relies not only on metadata about the activity, but also on metadata generated during the execution of the workflow.

Taverna is one of many scientific workflow management systems and supports what we might describe as "application level workflows", as opposed to some other systems which focus on scheduling tasks across computing resources. Taverna is used extensively by researchers in the life sciences. Its applications to date include gene and protein annotation, proteomics, phylogeny and phenotypical studies, microarray data analysis and medical image analysis, high throughput screening of chemical compounds and clinical statistical analysis. It is increasingly being adopted in other disciplines.

Two versions ot the Taverna software are currently available: SuperClient and Web Taverna. The SuperClient can be download and run on a researcher's desktop or laptop PC, without needing anything extra installed on servers or any system administration. Any services you can get at from your PC, whether they're in the enterprise or on the Web, can be plumbed together by running a Taverna workflow. The idea is that it should be like downloading and installing a Web browser to access information, except that it accesses remote services and runs workflows. Web Taverna is similar, except that it is installed on a Web server and invoked using a browser interface instead of being installed locally.

The Taverna Enactor: This is the engine that takes a Taverna workflow expressed in the SCUFL language and executes it over the services described within it and using the data provided by the user. In its early form the enactor did simple data staging from service to service, but more recently it supports streaming, determination of services from service groups at runtime and numerous extension points for developers.

The Taverna Workbench: This provides a graphical editor for workflows but also, significantly, the means for users to choose services – it is the availability of services for a particular domain that makes Taverna easy to use in that domain. It is easy to add new services. The workbench provides additional tools, for example it captures the execution logs of services in order to record the provenance of the results.

The Taverna Language: This is a simple data flow language called SCUFL (Simple Conceptual Unified Language) with implicit iteration constructs, manifest graphically in the Taverna editor and encoded in XML. It is actually a declarative language with its semantic roots in functional programming. It is quite different to languages such as BPEL that support a control flow paradigm.

XScufl is a simple workflow orchestration language for Web services which can handle WSDL based web service invocation. The main difference from BPEL is that XScufl, in association with a tool like Taverna allows programmers to write extension plug-ins, e.g. any kind of Java executable process, that can be used as part of the workflow.

6 Recommendations for Future Work

A sample implementation of a workflow driven application and subsequent evaluation could be done in WP3.4.2 if there is sufficient effort available. Some of this work will however be done in the new JISC funded NeISS project, see `http://www.neiss.org.uk`.

Table 1: Correspondence between SCUFL and BPEL4WS.

SCUFL	BPEL4WS	Xslt-rule	WOSE Service
stringconstant	Assign	Assign	
Arbitrarywsdl	Invoke	arbitraryInvoke	
Source	ReceiveData	Input	Getparameter
Sink	Reply	OutputResult	Tail service
Data link	Sequence activity	DatalinkAnalysis	
Concurrency constraint	Control activity	ControllinkAnalysis	

In general the different workflow tools use different description languages – this can be a pain. Considering the ones that use XML-based languages it might be possible to do a (partial) translation using XSLT.

The following are XML-based languages which might be of interest: AGWL, BPEL (BPEL4WS and WS-BPEL), WSFL, MoML, SCUFL, A-GWL, YAWL, GWDL,

Translation was investigated on a small scale in the WOSE project [1]. Table 1 shows the correspondence between SCUFL and BPEL4WS.

In general, all languages based on Web services share the same core model based on a directed graph, making transformation between them quite easy. (Actually P-GRADE is based on a Petri net.) Scientists should not need to learn different workflow description languages, and the generated workflow scripts could be re-used. XSLT is particularly useful for translation as it provides a high-level declarative programming language that can allow frequent changes to be made in the XML document describing the service. The XSLT converter is also beneficial in dealing with aspects beyond the current workflow languages such as workflow performance, QoS, etc. and transforming them into the implemental workflow scripts of publicly available workflow engines.

Consider the transformation from SCUFL to BPEL4WS (now called WS-BPEL) as an example. These are quite different workflow languages, but they share the same core model. The Simple Conceptual Unified Flow Language (SCUFL) is a high-level conceptual workflow language. The Business Process Execution Language for Web service (BPEL4WS) resulted from a merger of Microsoft's XLANG and IBM's WSFL, and is a block-structured programming language. Table 1 shows the correspondence between SCUFL and BPEL4WS elements. The XSLT-rules column shows the XSLT rules applied to transform from SCUFL to BPEL4WS. The Assign rule takes the value from a string constant processor in the SCUFL script and assigns it to a variable in the BPEL4WS script. The arbitraryInvoke rule takes the endpoint and operation from the SCUFL script and uses these as parameters for invoking Arbitrarywsdl Web service in the BPEL4WS script. The DataInput rule gets the parameters from an external parameter file by invoking the getparameter service. The OutputResult rule adds an XML display format tag into the results file. The DatalinkAnalysis rule analyses the sequence of activities and dataflow in SCUFL and generates the corresponding sequence of activities in BPEL4WS. The ControllinkAnalysis rule analyses the control structure of activities in SCUFL and generates the control structure of the corresponding activities in BPEL4WS.

Semantic enrichment of workflows has been discussed by Pignotti et al. [24] as a way of ensuring that they support the "scientists intent". This required the addition of rules expressed in SWRL with vocabulary and semantic descriptions using RDFS and OWL.

In summary, some of the technical challenges which must be addressed by the computer science community in order to make workflow systems usable by social scientists include the following.

- the underlying services in a workflow need to be provided with a standard interface, e.g. WS-I or WSRF;
- semantic descriptions need to be provided for such services so that they can be chosen and used as appropriate from a palette of available services;
- the notion of "similarity" of services needs to be considered, so that services can be replaced or updated;
- issues of service curation, provenance, validation and trustworthiness need to be addressed;
- there are many issues of access and security (particularly around data services) which need to be identified and addressed if workflows composed of many services are to be shared;
- the workflow plus its inputs and outputs should be capable of encapsulation along with semantic and packaging information as a "research object" which can be published;
- more than one workflow tool is in use and these can be installed on the user's desktop or on a server. Ideally workflows should be portable or a "broker" should be available;
- use of workflow alongside other tools in a Web based portal adds complications, for instance a portal based workflow editor is required;
- research workflows, e.g. ones using Grid resources, can be long running. Concepts of failover, rollback, resource leasing, monitoring, etc. need to be introduced. This has already been the focus of some research by the Grid community.

References

[1] R.J. Allan *Virtual Research Environments: from Portals to Science Gateways* (Chandos Publishing, Oxford, 2009) 230pp in press `http://www.woodheadpublishing.com/en/book.aspx?bookID=1892&ChandosTitle=1`

[2] R.J. Allan, R. Crouchley and C. Ingram *JISC Information Environment Portal Activity: Scenarios, Use Cases and Reference Models* (CCLRC, June 2006)

[3] R.J. Allan, R. Crouchley and C. Ingram *JISC Information Environment Portal Activity: Comparison of Surveys* (CSI Consultancy, June 2006)

[4] R.J. Allan, R. Crouchley and C. Ingram *JISC Information Environment Portal Activity: Final Report* (CCLRC, June 2006)

[5] T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, et al. *Business Process Execution Language for Web Services, Version 1.1* (2003) `ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf` `http://ifr.sap.com/bpel4ws.2003`

[6] D. Churches, G. Gombás, A. Harrison, J. Maassen, C. Robinson, M.S. Shields, I.J. Taylor and I. Wang *Programming Scientific and Distributed Workflow with Triana Services* Concurrency and Computation: Practice and Experience. 18:10 (Aug'2006) 1021-1037

[7] F. Curbera, R. Khalaf, W. Nagy and S. Weerawarana *Implementing BPEL4WS: the Architecture of a BPEL4WS Implementation* Concurrency and Computation: Practice and Experience. 18:10 (Aug'2006) 1219-28

[8] Y. Gil, V. Ratnakar, E. Deelman, G. Mehta and J. Kim *Wings for Pegasus: Creating Large Scale Scientific Applications Using Semantic Representations of Computational Workflows* In Proc. 19th Ann. Conf. on Innovative Applications of Artificial Intelligence (IAAI) (Vancouver, Canada, 2007)

[9] L. Huang, A. Akram, D.W. Walker, R.J. Allan, O.F. Rana and Y. Huang *A Workflow Portal Supporting Multi-Language Interoperation and Optimisation* Concurrency and Computation: Practice and Experience 19:12 (2007) 1583-95

[10] E.A. Lee and S. Neuendorffer *MoML – A Modeling Markup Language in XML – Version 0.4* Technical report (University of California at Berkeley, Mar'2000)

[11] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, A. Jaeger, M. Jones, E.A. Lee and J. Tao and Zhao, Y. *Scientific workflow management and the Kepler system* Concurrency and Computation: Practice and Experience. 18:10 (Aug'2006) 1039-65

[12] T. Oinn et al. *Taverna: Lessons in Creating a Workflow Environment for the Life Sciences.* Concurrency and Computation: Practice and Experience 18:10 (Aug'2006) 1067-1100

[13] E. Pignotti, P. Edwards, A. Preece, G. Polhill and N. Gotts *Semantic Workflow Management for e-Social Science* Paper presented at the 3rd International Conference on e-Social Science (2007)

[14] J.G. Polhill, E. Pignotti, N.M. Gotts, P. Edwards and A. Preece *A Semantic Grid Service for Experimentation with an Agent-Based Model of Land Use Change* Journal of Artificial Societies and Social Simulation 10:2 (2006) 2

[15] D. De Roure and C.A. Goble *Six Principles of Software Design to Empower Scientists* IEEE Software 26:1 (2009) 88-95 ISSN 0-740-7459 `http://eprints.ecs.soton.ac.uk/15032/`

[16] D. De Roure, C.A. Goble and R. Stevens *Designing the myExperiment Virtual Research Environment for the Social Sharing of Workflows.* In e-Science 2007 – Proc. 3rd IEEE International Conference on e-Science and Grid Computing Bangalore, India, (10-13/12/2007) 603-10

[17] I.J. Taylor, E. Deelman, D.B. Gannon and M. Sields (eds.) *Workflows for e-Science: Scientific Workflows for Grids* Springer (2007) 523pp ISBN 978-1-84628-519-6

[18] R.P. Tyer et al. *Remote My Condor Submit* NGS Wiki `http://wiki.ngs.ac.uk/index.php?title=Category:Community_Software`

[19] E. Bright Wilson *An Introduction to Scientific Research* (McGraw-Hill, 1952)

[20] *JISC Virtual Research Environments Programme* `http://www.jisc.ac.uk/index.cfm?name=programme_vre`

A Glossary, Abbreviations and URLs.

A glossary with many relevant entries can be found at: http://www.grids.ac.uk/ReDRESS/glossary_v2/glossary_v2.html.

Wikipedia can be used to obtain an explanation for most of the generic ones, http://www.wikipedia.org.

Specific abbreviations used in this report are:

BPEL: Business Process Execution Language

BPEL4WS: BPEL for Web Service V1.1 http://www.ibm.com/developerworks/library/specification/ws-bpel/

BPML: Business Process Modelling Language

CCLRC: Council for the Central Laboratory of the Research Councils, now part of STFC

Condor: Middleware for Campus Grids. Includes DAGMan which is a workflow management tool.

CQeSS: Collaboratory for Quantitative e-Social Science (NCeSS Node)

DAG: Directed Acyclic Graph

eReSS: JISC e-Research Interoperability and Standards project

EDINA: is not an acronym, but the old poetic name for Edinburgh as use by Robert Burns. EDINA is the JISC national academic data centre based at University of Edinburgh http://edina.ac.uk

ESRC: Economic and Social Research Council http://www.esrc.ac.uk

GEMS: Grid Enabled MIMAS Service, JISC-funded project http://pascal.mvc.mcc.ac.uk:9080/gems

GIS: Geographical Information System

GLS: Geographical Linking Service

Grid: A collection of distributed computing and data resources connected by middleware.

GROWL: Grid Resources on Workstation Library, JISC-funded VRE-1 project http://www.growl.org.uk

HTTP: Hyper-Text Transport Protocol

IEMSR: Information Environment Meta-Data Schema Registry http://iemsr.ac.uk

IESR: Information Environment Service Registry http://iesr.ac.uk

Jini: Java services based middleware

JXTA: Java peer-to-peer middleware

Kepler: Workflow system from SDSC

Kuali: Workflow system from acadmic adminstration processes

MoML: Modelling Markup Language `http://ptolemy.eecs.berkeley.edu/projects/summaries/00/moml.html`

MOSeS: Modelling and Simulation for e-Social Science (NCeSS Node)

NCeSS: National Centre for e-Social Science `http://www.ncess.ac.uk`

NeISS: National e-Infrastructure for Social Simulation `http://www.neiss.org.uk`

NGS: National Grid Service `http://www.ngs.ac.uk`

OGSA: Open Grid Services Architecture

OGSA-DAI: Data Access and Integration using Open Grid Services `http://www.ogsadai.org.uk`

P2P: Peer-to-peer, see Wikipedia

Pegasus: Workflow system from the GryPhyn project

Petri Net: Used for graph-based modelling of distributed systems, e.g. in the P-GRADE workflow system.

RCUK: Research Councils UK `http://www.rcuk.ac.uk`

ReDReSS: Resource Discovery for Researchers in e-Social Science

RMCS: Remote MyCondor Submit workflow system from eMinerals

SCUFL: Simple Conceptual Unified Flow Language used in Taverna. See XSCUFL `http://www.ebi.ac.uk/tmo/mygrid/XScuflSpecification.html`

REST: a non-standard approach to accessing remote services using HTTP's PUT, GET and POST rather than WS-I

SOA: Service Oriented Architecture

SOAP: Originally stood for Simple Object Access Protocol, the basis of Web services

SRB: Storage Resource Broker `http://www.npaci.edu/DICE/SRB/`

STFC: Science and Technology Facilities Council, formed by combining CCLRC and PPARC, see `http://www.stfc.ac.uk`

Taverna: Workflow system from MyGrid

Triana: Workflow system from University of Cardiff.

VO: Virtual Organisation

WSDL: Web Service Desciption Language

WSRF: Web Service Resource Framework

Web Services (WS-I): Language agnostic remote method invocation using XML, SOAP, WSDL and UDDI

UDDI: Universal Description, Discovery and Integration, a Web services registry specification, see Wikipedia

B Some Scientific Workflow Tools

The original source of this information is http://www.extreme.indiana.edu/swf-survey/. This has been partially updated. The original site has links to more information about each tool.

This can now be found on Wikipedia http://wiki.cogkit.org/index.php/Scientific_Workflow_Survey.

Name	Status	Pre-requisites	Tooling	Standards	Grid integration	Portal integration	Comments
Askalon	N/A	Java SDK 1.4, CoG	Java GUI	Custom XML (AGWL)	GT3, CoG	N/A	A Grid runtime for AGWL that has resource broker, resource monitoring, (meta)-scheduler etc.
AGWL	N/A	N/A	N/A	N/A	N/A	N/A	AGWL is an abstract Grid workflow language for describing workflows (graphs with loops) at a high level of abstraction used in Askalon
BioOpera							
Chimera		Condor DAG-Man	?	N/A	Condor	?	Built on top of Condor, workflow as solution to provide on-demand data generation ("virtual data")
D2K	Multiple License	JDK 1.3.1	Java GUI	N/A	?	?	Modules composed into data flow optimized for text and data mining and KDD (Knowledge Discovery in Databases)
DAGMan	GPL	Condor	Integrated with Condor command line tools	N/A	Condor, can run on top of GT2 (Condor-G)	Under work	Part of Condor, very well integrated
DDBJ							
DiscoveryNet							
EUROGRID							

Fiswidgets	GPL	Java (version?)	Java GUI	custom XML	not yet	no	domain independent, but designed for processing neuroimaging analyses
GRID superscalar	Binary on request (license?)				GT2		Simple makefile like data dependency to describe and execute grid tasks
GridAnt	Soon?	Java JDK 1.4, CoG	Graphical workflow visualizer (ANL)	Built on top of Apache ANT	via CoG	N/A	Merged ANL GridAnt and NCSA OGRE
GridPort	3.0 alpha	J2EE, CoG, JBoss	N/A	NA	via CoG	NMI OGCE, Jetspeed, Hotpage 3.0	Non standard description of job sequences
GridNexus	1.0.0 (OSS license planned)	Java JDK 1.4	Built on top of PtolemyII Java GUI	Internal DAG is transformed to new scripting language called JXPL (XML based)	GT3	Under investigation	Main focus on making Drag-and-drop Grid GUI
Grid Service Broker	Part of the Gridbus Broker, code available under GPL	Java, CoG	Command line, Java based Grid API,		GT2, Alchemi	Yes, with G-monitor that supports multiple devices (?)	Supports parametric computing for compute and data grids applications. being extended to support advanced workflows.
GRMS	GPL	Java, CoG	command line and GridSphere Web portlet	DAG/ PetriNet	Pre-WS GT	yes (GridSphere)	GRMS is a persistent GSI-enabled Web Service in GridLab project
GSFL	NA (only paper)						Extended WSFL to add Grid support

JOpera for Eclipse	BSD	Java 1.4 (or above)	Eclipse 3.1	Directed Cyclic Labeled Graphs, stored in custom XML format (Graph Flow Diagrams) - supports cycles	WSDL, WSRF, GT4, SSH, Condor	Built-in monitoring tools (Eclipse RCP and Web based)	Grid workflows fully integrated with the Eclipse user experience; Extensible with your own plugins to call any kind of service
ICENI	SISSL (open source license - certified?)	Java	Java GUI				Component based dataflow
I-Lab	Soon?	Java?	Java GUI (Visual Grid Job Authoring)	GADL is based on Petri Nets	GT2	Job Builder	Allows graph refinement during execution (such as add file transfer node)
INFORMNET	Summer'04?	Java?	Visualizer?	NA	GT3	?	uses XML based DAG
Karajan	CoG2 CVS see CoG2 Manual for details)	Java 1.4, CoG2	command line, API, GUI?	Ant like	GT2.4, GT3.02, SSH (future targets: Condor, GT4.0, Unicore)	NMI OGCE.org	Aims to be as easy to use as ANT but more workflow oriented
Pegasus	?	Condor DAGMan			Condor		Pegaus translates Abstract Workflow (AW) to produce a Concrete Workflow (CW) submitted to Condor's DAGMan
Kepler/ PtolemyII	Source under Berkeley License	Java JDK 1.4, WSIF	Java GUI	N/A	Under work	No	Uses the MoML XML language to represent workflow
P-GRADE	?	C (LINUX, Solaris, IRIX)	GUI	NA	GT2, Condor	Supported	P-GRADE Workflow Language

SCIRun	Available under University of Utah Public License (not free for commercial use - issues raised in GNU)	C++ on Linux or IRIX			SDSC SRB, CoG	N/A	Specializes in dataflow visualization
SDSC Matrix Project	Apache style	Java JDK 14, CoG, servlet container (Tomcat)	None	N/A	CoG		Data Grid Language (DGL)
SWFL	N/A						Service Workflow Language
Taverna (part of MyGrid)	Source code under LGPL license	Java JDK 1.4	multiple projects: Freefluo - workflow enactment engine, Taverna Workbench - GUI designer	N/A	?	via Talisman toolkit that allows scriptable Web UI (not active)	new XML language called Scufl that is built to support scientific applications
Teutaa	?	Java JDK 1.4	GUI Tool to build UML model	UML Activity Diagrams	?	?	UML diagram is mapped to Abstract Grid Workflow Language (A-GWL) and then to concrete C-GWL
Triana	source code under Apache License	Java	Java GUI				Part of GridLab
Unicore							
wftk							

B.1 Links to Further Information

ActiveBPEL `http://www.activeBPEL.org/`

Askalon `http://dps.uibk.ac.at/projects/askalon/`

AGWL `http://dps.uibk.ac.at/projects/agwl/`

BioOpera `http://www.extreme.indiana.edu/swf-survey/BioOpera.html`

Chimera `http://www.extreme.indiana.edu/swf-survey/Chimera.html`

D2K `http://alg.ncsa.uiuc.edu/do/tools/d2k`

DAGMan `http://www.extreme.indiana.edu/swf-survey/DAGMan.html`

DDBJ `http://www.extreme.indiana.edu/swf-survey/DDBJ.html`

EUROGRID `http://www.extreme.indiana.edu/swf-survey/EUROGRID.html`

Fiswidgets `http://grommit.lrdc.pitt.edu/fiswidgets/`

DiscoveryNet `http://www.extreme.indiana.edu/swf-survey/DiscoveryNet.html`

GridAnt`http://www.extreme.indiana.edu/swf-survey/GridAnt.html`

GridPort `http://www.extreme.indiana.edu/swf-survey/GridPort.html`

GridNexus `http://www.extreme.indiana.edu/swf-survey/GridNexus.html`

Grid Service Broker `http://www.extreme.indiana.edu/swf-survey/GridServiceBroker.html`

Alchemi `http://www.alchemi.net/`

GRMS `http://www.gridlab.org/grms`

GSFL `http://www.extreme.indiana.edu/swf-survey/GSFL.html`

JOpera for Eclipse `http://www.extreme.indiana.edu/swf-survey/JOpera.html`

ICENI `http://www.extreme.indiana.edu/swf-survey/ICENI.html`

I-Lab `http://www.extreme.indiana.edu/swf-survey/ILab.html`

INFORMNET `http://www.extreme.indiana.edu/swf-survey/INFORMNET.html`

Karajan `http://www.extreme.indiana.edu/swf-survey/Karajan.html`

Pegasus `http://www.extreme.indiana.edu/swf-survey/Pegasus.html`

Kepler `http://www.extreme.indiana.edu/swf-survey/Kepler.html`

PtolemyII `http://www.extreme.indiana.edu/swf-survey/PtolemyII.html`

P-GRADE `http://www.extreme.indiana.edu/swf-survey/P-GRADE.html`

SDSC Matrix Project `http://www.extreme.indiana.edu/swf-survey/SDSCMatrix.htm`

SWFL `http://www.extreme.indiana.edu/swf-survey/SWFL.html`

Taverna (part of MyGrid) `http://www.extreme.indiana.edu/swf-survey/Taverna.html` and `http://taverna.sourceforge.net/`

Teutaa `http://www.extreme.indiana.edu/swf-survey/Teuta.htm`

Triana `http://www.extreme.indiana.edu/swf-survey/Triana.html`

Unicore `http://www.extreme.indiana.edu/swf-survey/Unicore.html`

wftk `http://www.extreme.indiana.edu/swf-survey/wftk.html`

WSFL `http://www-306.ibm.com/software/solutions/webservices/pdf/WSFL.pdf`

XLANG `http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm`

XSLT `http://www.w3.org/TR/xslt`