Semantic Workflow Management for E-Social Science

E. Pignotti[1], P. Edwards[1], A. Preece[1], G. Polhill[2], N. Gotts[2]

[1]Dept. of Computing Science, University of Aberdeen.
[2]The Macaulay Institute, Craigiebuckler, Aberdeen.

e.pignotti@abdn.ac.uk

Abstract. In the e-Science context, workflow technologies provide a problem-solving environment for researchers by facilitating the creation and execution of experiments from a pool of available data and computation services. Recent activities in the field of social simulation indicate the need to improve the scientific rigour of agent-based modelling by making simulation experiments more transparent. We argue that in order to characterise such experiments we need to go beyond low-level service composition and execution details by capturing higher-level descriptions of the scientific process making the experiment's constraints and goals transparent. Current workflow technologies do not incorporate any representation of these goals and conditions, which we call the *scientist's intent*. Our hypothesis is that by extending workflow representation in this way, researchers would be able to analyse, verify, execute, monitor and re-use scientific workflows more effectively.

Introduction

In recent years there has been a proliferation of scientific resources available through the internet such as datasets and computational modelling services. Scientists are becoming more and more dependent on these resources, which are changing the way they conduct their day to day research activity (with increasing emphasis on "*in silico*" experiments as a computational means to test a hypothesis). Scientific workflow technologies (Pennington, D. 2007) have emerged as a problem-solving environment for researchers by facilitating the creation and execution of experiments given a pool of available data and computation services.

As part of the PolicyGrid[1] project we 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. The project involves collaboration between computer scientists and social scientists at the University of Aberdeen, the Macaulay Institute (Aberdeen) and elsewhere in the UK. The project aims to support policy-related research activities within social science by developing appropriate Grid middleware tools which meet the requirements of social science practitioners. The project is developing a range of services to support social scientists with mixed-method data analysis (involving both qualitative and quantitative data sources) together with the use of social simulation techniques. Issues surrounding usability of Semantic Grid tools are also a key

[1] Funded under the UK Economic and Social Research Council e-Social Science programme; grant reference RES-149-25-1027 (http://www.policygrid.org)

feature of PolicyGrid, with activities encompassing workflow support and natural language presentation of metadata.

The main benefit of current workflow technologies is that they provide a user-friendly environment for both the design and enactment of experiments without the need for researchers to learn how to program. Many different workflow languages exists including: MoML[2] (Modelling Markup Language), BPEL[3] (Business Process Execution Language), Scufl[4] (Simple conceptual unified flow language). All these languages are designed to capture the flow of information between services (e.g. service addresses and relations between inputs and outputs).

A typical experimental research activity (Wilson, E. Bright. 1952) involves the following steps: observation, hypothesis, prediction (under specified constraints), experiment, analysis and write-up. While workflow technologies provide support for a researcher to define an experiment, there is no support for capturing the constraints associated with it, therefore making it difficult to situate the experiment in context, We argue 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,* transparent.

Motivation & Example

Recent activities in the field of social simulation (Polhill et. al., 2006) 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. Yet results gathered from possibly hundreds of thousands of simulation runs cannot 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. We have identified activities that are relevant to such situations. These are:

- 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.

In a previous project, FEARLUS-G (Pignotti, et al. 2005), we tried to meet the needs of agent based modelling using Semantic Grid technologies (De Roure, et al. 2001). FEARLUS-G aimed to provide scientists interested in land-use phenomena with a means to run much larger-scale experiments than is possible on standalone PCs, and also gave them a Web-based environment in which to share simulation results. The FEARLUS-G project developed an ontology which centred on the tasks and entities involved in simulation work, such as

[2] http://ptolemy.eecs.berkeley.edu/projects/summaries/00/moml.html

[3] http://www.ibm.com/developerworks/library/ws-bpel/

[4] http://www.cs.man.ac.uk/~witherd5/taverna-site/scufl/index.html

experiments, hypotheses, parameters, simulation runs, and statistical procedures. In FEARLUS-G we showed that it is possible to capture the context in which a simulation experiment is performed making collaboration between scientists easier. However, FEARLUS-G was not designed to be a flexible problem-solving environment as the experimental methodology was hard-coded into the system. We feel that, in this context, workflow technologies can facilitate the design, execution, analysis and interpretation of simulation experiments and exploratory studies. However, we argue that current workflow technologies can only capture the method and not the scientist's intent which we feel is essential to make the experiment truly transparent.

We have identified a number of scenarios through interaction with collaborators from the social simulation community. We now present a simulation case study using a virus model developed in NetLogo[5]: an agent-based model that simulates the transmission and perpetuation of a virus in a human population. An experiment using this model might involve studying the differences between different types of virus in a specific environment. A researcher wishing to test the hypothesis 'Smallpox is more infectious than bird flu in environment A' might run a set of simulations using different random seeds. If in this set of simulations, *Smallpox* outperformed *Bird Flu* in a significant number of simulation runs, the experimental results could be used to support the hypothesis.

Figure 1 (bottom) shows a workflow built using the Kepler editor tool (Ludäscher et al. 2005) that uses available services to perform the experiment described above. The *VirusSimulationModel* generates simulation results based on a set of parameters loaded at input from a data repository; the experiment definition is selected by *Experiment ID.* These simulation results are aggregated and fed into the *Significance Test* component which outputs the results of the test. The hypothesis is tested by looking at the result of the significance test; if one virus that we are considering (e.g. *smallpox*) significantly outperforms another, we can use this result to support our hypothesis.

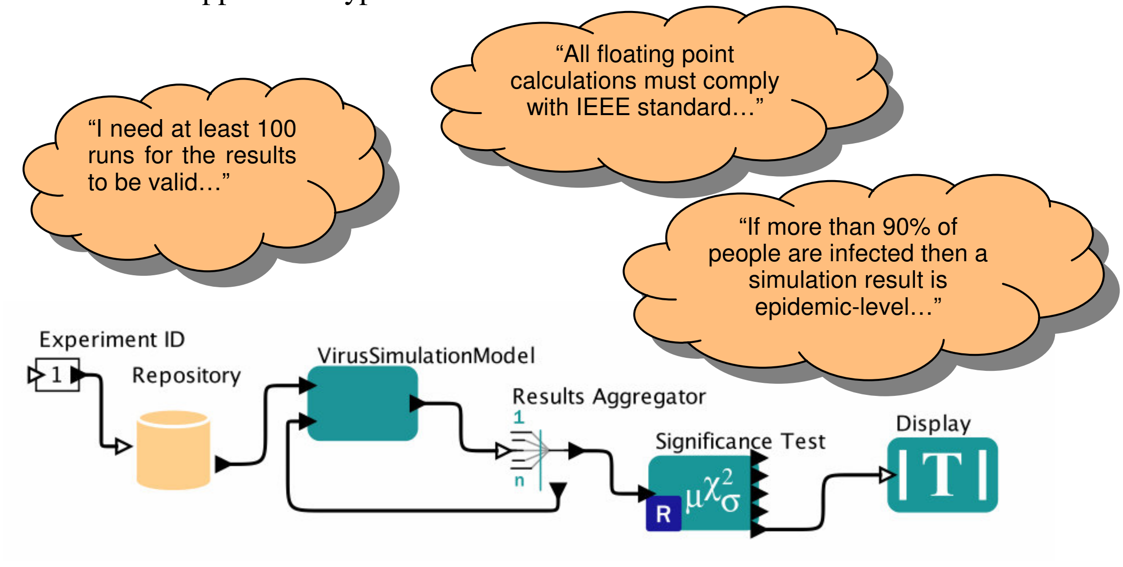

"All floating point calculations must comply with IEEE standard…"
"I need at least 100 runs for the results to be valid…"
"If more than 90% of people are infected then a simulation result is epidemic-level…"
Experiment ID
1
Repository
VirusSimulationModel
Results Aggregator
1
n
Significance Test
R
$\mu\chi^2_\sigma$
Display
|T|

Figure 1 - Simulation Workflow Example (bottom) & Scientist's Intent (top).

The experimental workflow outlined in Figure 1 has some limitations as it is not able to capture the scientist's goals and conditions (*scientist's intent*) as illustrated in Figure 1 (top). For example, the goal of this experiment is to obtain significant simulation results that support

[5] http://ccl.northwestern.edu/netlogo/

the hypothesis. Imagine that the researcher knows that the simulation model could generate out-of-bounds results and these results cannot be used in the significance test. For this reason, we don't know *a priori* how many simulation runs per comparison we need to do. Too few runs will mean that the experiment will return inconclusive data, while too many runs will waste computing resources executing unnecessary simulations. There may also be constraints associated with the workflow (or specific activities within the workflow) depending upon the intent of the scientist. For example, a researcher may be concerned about floating point support on different operating systems; if the *Significance Test* activity runs on a platform not compatible with IEEE 754 specifications, the results of the simulation could be compromised. A researcher might be also interested in detecting and recording special conditions (e.g. a particularly virulent virus) during the execution of the workflow to support the analysis of the results. Existing workflow languages are unable to explicitly associate such information with their workflow descriptions.

The main challenges we face are to represent *scientists intent* in such a way that:

- it is meaningful to the researcher, e.g. providing information about the context in which an experiment has been conducted so that the results can be interpreted;
- it can be reasoned about by a software application, e.g. an application can make use of the intent information to control, monitor or annotate the execution of a workflow;
- it can be re-used across different workflows, e.g. the same high-level intent may apply to different workflows;
- it can be used as provenance (documenting the process that led to some result).

Related Research

Many of the concepts underlying today's eScience workflow technologies originated from business workflows. These typically describe the automation of a business process, usually related to a flow of documents. Scientific workflow on the other hand is about the composition of structured activities (e.g. database queries, simulations, data analysis activities, etc.) that arise in scientific problem solving (Ludäscher et al. 2005). However, the underlying representation of the workflow remains the same (data and control flow). For example the language BPEL (Andrews et al. 2003), originally designed for business, has been adapted for scientific workflow use. BPEL4WS[6] 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).

XScufl[7] 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

[6] http://www.ibm.com/developerworks/library/specification/ws-bpel/

[7] http://www.ebi.ac.uk/~tmo/mygrid/XScuflSpecification.html

association with a tool like Taverna (Oinn et al. 2004) allows programmers to write extension plug-ins (e.g. any kind of Java executable process) that can be used as part of the workflow. Taverna is a tool developed by the myGrid 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.

MoML (Lee and Neuendorffer 2000) 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 (Ludäscher et al. 2005), is a workflow tool based on the MoML language and Ptolemy-II system for heterogeneous, concurrent modeling and design. Kepler allows "Drag&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 (due to 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 our framework can operate.

Gil et al (2007) 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. Although in our research we are not focusing on assisted workflow composition, we do share the same interest in the benefit of enhanced semantics in workflow representation. While both our approaches rely on logical statements that apply to workflow metadata, we 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.

Approach

As part of our approach we are proposing a framework for capturing the scientist's intent (based upon rules) so the formal representation of the intent can be used to reason about the workflow. The intent rules will operate on metadata generated by a workflow. Details of the intent are kept separate from the operational workflow as embedding constraints and goals directly into the workflow representation would make it overly complex (e.g. with a large number of conditionals) and would limit potential for sharing and re-use. Such a workflow would be fit for only one purpose and addition of new constraints would require it to be substantially re-engineered. Using the support for scientific intent, a new experiment might be created just by changing the rules but not the underlying operational workflow.

In this section we present a semantic workflow infrastructure solution based on scientist's intent, highlighting the requirements for the various components. We base our solution on open workflow frameworks (e.g. Kepler) that allow the creation and execution of workflows based on local, Grid or Web services. A key part of this infrastructure is the workflow metadata support which provides information about the workflow components, inputs and outputs and their execution. We also require a scientist's intent framework that captures goals and prerequisites of the experiment based on the workflow metadata.

Open Workflow Frameworks

Open workflow frameworks are the core of our solution as they provide the tools and systems to model and execute workflow. Different workflow frameworks may take different approaches; in this section we highlight the core functionality necessary to provide support for our solution. An important element of a workflow framework is the modelling tool (or editor) that allows researchers to design a workflow from available services. The key requirement here is that the editor is capable of working with both local and Grid services and that the resulting workflow is represented in a portable and machine processable language (e.g. XML). Workflow frameworks also provide the execution environment necessary to enact the workflow. Usually the execution environment provides a monitoring tool which allows the scientist to inspect the status of the execution. An important requirement is the ability to monitor and control the workflow execution through the use of APIs from external applications. This will provide the appropriate software in which the scientist's intent framework can operate.

Workflow Metadata Support

A crucial aspect of our framework is that the workflow must have supporting ontologies and should produce metadata that can be used against scientific intent to "reason" about the workflow. We have identified the following possible sources of metadata:

- metadata about the result(s) generated upon completion of the workflow (e.g. a significance test);
- metadata about the data generated at the end of an activity within the workflow or sub-workflow (e.g. simulation model run);
- metadata about the status of an activity over time, for example while the workflow is running (e.g. infected people, immune people).

In order to support social simulation activities within our framework we have created an initial social simulation classification ontology[8] capturing the characteristics of a wide range of simulation models (e.g *typeOfSimulationModel*, *executionModel*, *typeOfModelBehavior*). We are also continuing work on the development of a simulation modelling ontology to allow a particular piece of modelling software to be described and the structure and context of a

[8] http://www.csd.abdn.ac.uk/research/policygrid/ontologies/SocialSimulationOntology.owl

particular simulation run to be characterised. Other ontologies will be created in due course to fit different case-studies.

Grid & Web Services

Central to our idea of capturing intent is the concept of the Semantic Grid. Where Grid technologies (Foster and Kesselman 2001) provide an infrastructure to manage distributed computational resources (e.g. *VirusSimulationModel*), the vision of the Semantic Grid is based upon the adoption of metadata and ontologies to describe resources (services and data sources) in order to promote enhanced forms of collaboration among the research community. Ontologies are used to capture the meaning of metadata terms and their interrelationships. The main benefit of using ontologies is that they facilitate access to heterogeneous and distributed information sources by defining a machine-processable semantics for those information sources. Moreover, ontologies facilitate intelligent search mechanisms and automated reasoning services. Important technologies include RDF Schema (RDFS)[9] - a vocabulary for describing properties and classes of RDF resources, with semantics for generalization-hierarchies; and OWL (Web Ontology Language)[10] – which adds more vocabulary for describing properties and classes, e.g. relations between classes, cardinality, etc. The main benefit of using metadata enriched resources is that they provide supporting information so that shared and precisely defined terms (e.g. *virus*, *experiment*, *simulation model*, *floating point standard*, etc.) can be used in the context of scientist's intent.

Rules

We have identified SWRL[11] (Semantic Web Rule Language) as a language for capturing rules associated with *scientist's intent*. SWRL enables Horn-like rules to be combined with metadata. The rules take the form of an implication between an antecedent (body) and consequent (head). The intended meaning can be read as: whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold. This formalism suitable for capturing scientist's intent, as the rules capture the logic behind the intent while the ontology and metadata about the workflow provides the "knowledge base" upon which the rules can operate.

The scientific intent reflected in the example in Figure 1 can be represented as a combination of goals and constraints. For example:

Goal: Run enough simulations to provide valid results to support
(valid-run > 100)

Constraints: Significance Test has to run on a platform compatible with IEEE 754
(platform = IEEE 754).

Figures 2 and 3 show examples of intent rules based on the workflow presented in Figure 1. We will now discuss each of these in turn.

[9] http://www.w3.org/TR/rdf-schema/

[10] http://www.w3c.org/2004/OWL/

[11] http://www.w3.org/Submission/SWRL/

The rule in Figure 2 is used to identify if the significance test activity is running on a platform compatible with IEEE 754.

```
      platform  (?x1,”IEEE754”)  ^
     hasResult  (?x1,?x2)
                       =>
hasValidresult  (?x1,?x2)
```

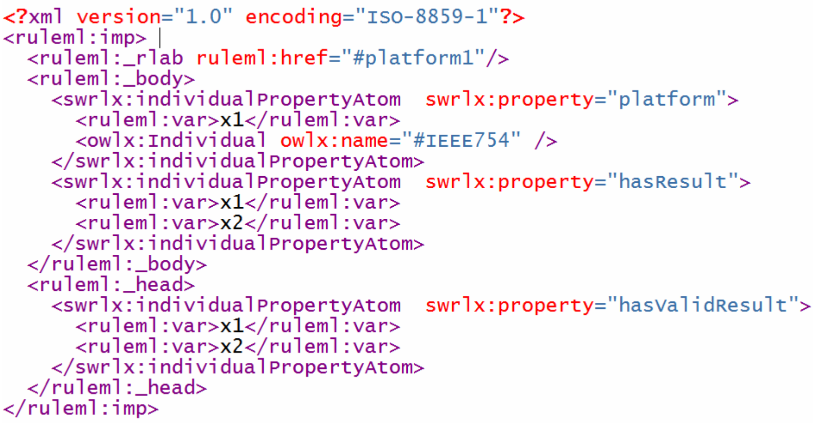
Figure 2 – Example Rule: Significance Test.

While the previous rule supports the verification and execution of the workflow by identifying invalid results or simulation runs, the rule in Figure 3 aims to facilitate the analysis of results by enriching them with additional metadata. For example: *if the number of infected people in a simulation run is more than 90%, the virus tested is epidemic.*

```
          virus  (?x1)          ^
     virusModel  (?x2)          ^
      testVirus  (?x2,?x1)      ^
    hasModelRun  (?x2,?x3)      ^
 infectedPeople  (?x3,?x4)      ^
      more-than  (?x4,90%)
                      =>
isEpidemicVirus  (?x1)
```

Figure 3 - Example Rule: Semantic Enrichment.

Figure 4 shows the underlying SWRL representation based on a supporting OWL ontology[12] and represented using XML syntax. We decided to adopt this solution because it fits very well in terms of integration with existing workflow metadata support.



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ruleml:imp>
  <ruleml:_rlab ruleml:href="#platform1"/>
  <ruleml:_body>
    <swrlx:individualPropertyAtom  swrlx:property="platform">
      <ruleml:var>x1</ruleml:var>
      <owlx:Individual owlx:name="#IEEE754" />
    </swrlx:individualPropertyAtom>
    <swrlx:individualPropertyAtom  swrlx:property="hasResult">
      <ruleml:var>x1</ruleml:var>
      <ruleml:var>x2</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_body>
  <ruleml:_head>
    <swrlx:individualPropertyAtom  swrlx:property="hasValidResult">
      <ruleml:var>x1</ruleml:var>
      <ruleml:var>x2</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_head>
</ruleml:imp>
```

Figure 4 - SWRL Rule Representation.

Actions based on scientist’s intent (e.g. **`IF hasInvalidRun(?x1,?x2) THEN ignore(?x2)`**) will depend on the ability of the workflow framework to detect events from the scientist’s intent framework and execute an action. We are currently extending the Kepler

[12] http://www.w3.org/Submission/SWRL/swrl.owl

workflow tool to operate with our scientist's intent framework by registering the events that it is capable to detect and perform.

Evaluation & Conclusions

Our evaluation strategy involves assessing the usability of the enhanced workflow representation using real workflows from the case-studies identified with our collaborators. We are using Kepler as a design tool and Grid services that we have developed over time as workflow activities (e.g. various simulation models). User scientists are central to the evaluation process, as they will use the tools and then supply different types of feedback via questionnaire, interview or through direct observation.

Lack of space prevents us discussing the evaluation plan in detail, but we will now present our key evaluation criteria:

- **Expressiveness of the intent formalism:** Is the formalism sufficient to capture real examples of intent? Were certain constraints impossible to express? Were some constraints difficult to express?
- **Reusability**: Can an intent definition be reused - either in its entirety or in fragments? Does our framework facilitate reusability?
- **Workflow execution:** Does the inclusion of intent information affect the computational resources required during the execution of a workflow? (This type of evaluation will be carried out in simulated conditions by running workflow samples with and without scientist's intent support and measuring the Grid resources used and the time involved.)

From a user perspective, creating and utilizing metadata is a non-trivial task; the use of a rule language to capture scientists' intent does of course provide additional challenges in this regard. Although we have not currently addressed these issues in this research, other work ongoing within the PolicyGrid project may provide a possible solution. Hielkema et al. 2007 describe a tool which provides access to metadata (create, browse and query) using natural language. The tool can operate with different underlying ontologies, and we are sure that it could be extended to work with SWRL rules - creating a natural language interface for defining and exploring scientist's intent.

In conclusion, we aim to provide a closer connection between experimental workflows and the goals and constraints of the researcher, thus making experiments more transparent. While the scientist's intent provides context for the experiment, its use should also facilitate improved management of workflow execution.

Acknowledgments

This research is funded by the Aberdeen Centre for Environmental Sustainability (ACES) and conducted within the PolicyGrid project involving the Macaulay Institute and the University of Aberdeen; PolicyGrid is funded by the UK Economic & Social Research Council, award reference RES-149-25-1027.

References

Andrews, T. (2003): *Business Process Execution Language for Web Services, Version 1.1* ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf, 2003.

De Roure, D., Jennings, N. R., Shadbolt, N. R. (2001): *Research Agenda for the Semantic Grid: A Future e-Science Infrastructure*, National e-Science Centre, Edinburgh, UK UKeS-2002-02, December 2001.

Foster, I., Kesselman, C. (1998): *Globus: A Toolkit-Based Grid Architecture*, In: The Grid: Blueprint for a Future Computing Infrastructure. Morgan- Kaufmann, 1998, pp. 259–278.

Gil, Y., Ratnakar, V., Deelman, E., Mehta, G., Kim, J. (2007): *Wings for Pegasus: Creating Large-Scale Scientific Applications Using Semantic Representations of Computational Workflows,* In Proceedings of the 19th Annual Conference on Innovative Applications of Artificial Intelligence (IAAI), Vancouver, British Columbia, Canada, July 22-26, 2007.

Hielkema, F., Edwards, P., Mellish, C., Farrington, j. (2007): *A Flexible Interface to Community-Driven Metadata,* To appear in Proceedings of the e Social Science conference 2007, Ann Arbor, Michigan 2007.

Lee, E. A., Neuendorffer, S. (2000): *MoML — A Modeling Markup Language in XML — Version 0.4*, Technical report, University of California at Berkeley, March, 2000.

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

Neuendorffer, L. (2000): *MoML - A Modeling Markup Language in XML - Version 0.4*

Oinn, T., Addis, M., Ferris, J. Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M.R., Wipat, A., Li, P.. (2004): *Taverna: A Tool for the Composition and Enactment of Bioinformatics Workflows*, In: Bioinformatics Journal 20(17) pp 3045-3054, doi:10.1093/bioinformatics/bth361.

Pennington, D. (2007): *Supporting Large-Scale Science with Workflows*, In: Proceedings of the 2nd workshop on Workflows in support of large-scale science, High Performance Distributed Computing 2007.

Pignotti, E., Edwards, P., Preece, A., Polhill, G., Gotts, N. (2005): *Semantic Support for Computational Land-Use Modelling*, Proceedings of the Fifth IEEE International Symposium on Cluster Computing and Grid (CCGrid)2005, IEEE Press, 2005, vol 2, pp 840-847.

Polhill, J. G., Pignotti, E., Gotts, N. M., Edwards, P. and Preece, A. (2007). 'A Semantic Grid Service for Experimentation with an Agent-Based Model of Land-Use Change'. Journal of Artificial Societies and Social Simulation 10(2)2.

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