Enhancing Workflow with a Semantic Description of Scientific Intent

Edoardo Pignotti[1], Peter Edwards[1], Alun Preece[2], Nick Gotts[3], and Gary Polhill[3]

[1] School of Natural & Computing Sciences , University of Aberdeen
Aberdeen, AB24 5UE, UK
{e.pignotti,p.edwards}@abdn.ac.uk
[2] School of Computer Science, Cardiff University
Cardiff, CF24 3AA, UK
A.D.preece@cs.cf.ac.uk
[3] The Macaulay Institute
Craigiebuckler, Aberdeen, AB15 8QH, UK
{n.gotts,g.polhill}@macaulay.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 services. In this paper we will show how Semantic Web technologies can be used to overcome a limitation of current workflow languages by capturing experimental constraints and goals, which we term *scientist's intent*. We propose an ontology driven framework for capturing such intent based on workflow metadata combined with SWRL rules. Through the use of an example we will present the key benefits of the proposed framework in terms of enriching workflow output, assisting workflow execution and provenance support. We conclude with a discussion of the issues arising from application of this approach to the domain of social simulation.

Keywords: eScience, semantic grid, workflow, SWRL, constraints, goals.

1 Introduction

In recent years there has been a proliferation of scientific resources available through the internet, including datasets and computational modelling services. Scientists are becoming more and more dependent on these resources, which are changing the way they conduct their research activities (with increasing emphasis on 'in silico' experiments as a computational means to test a hypothesis). Scientific workflow technologies [1] have emerged as a problem-solving tool for researchers by facilitating the creation and execution of experiments given a pool of available 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

[1] http://www.policygrid.org

S. Bechhofer et al.(Eds.): ESWC 2008, LNCS 5021, pp. 644–658, 2008.

UK. The project aims to support policy-related research activities within social science by developing appropriate Semantic Grid [2] tools which meet the requirements of social science practitioners. Where Grid technologies [3] provide an infrastructure to manage distributed computational resources, 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. The PolicyGrid 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 tools are also a key feature of PolicyGrid, with activities encompassing workflow support and natural language presentation of metadata [4].

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 exist including: MoML (Modelling Markup Language) [5], BPEL (Business Process Execution Language) [6], Scufl (Simple conceptual unified flow language) [7]. All these languages are designed to capture the flow of information between services (e.g. service addresses and relations between inputs and outputs).

As more computational and data services become available and researchers begin to share their workflows and results, there will be an increasing need to capture provenance associated with such workflows. Provenance (also referred to as lineage or heritage) aims to provide additional documentation about the processes that lead to some resource [8]. Goble [9] expands on the Zachman Framework [10] by presenting the '7 W's of Provenance': *Who, What, Where, Why, When, Which, & (W)How*. While some progress has been made in terms of documenting processes [11] (*Who, What, Where, When, Which, & (W)How*), little effort has been devoted to the *Why* aspect of research methodology. This is particularly important in the context of policy appraisal [12] .

A typical experimental research activity [13] 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 conditions under which the experiment is conducted, therefore making it difficult to situate the experiment in context. While existing provenance frameworks can provide information about an experiment by documenting the process, we argue that in order to fully characterise scientific analysis we need to go beyond such low-level descriptions by capturing the experimental conditions. The aim here is to make the constraints and goals of the experiment, which we describe as the *scientist's intent*, transparent.

PolicyGrid aims to provide an appropriate provenance framework to support evidence-based policy assessment where the focus is on how a particular piece of evidence was derived. To date the project has developed a `resource`[2] and a `task`[3] ontology to capture such provenance information. The `resource` ontology describes the type of resources used by social scientists (e.g. `Questionnaire`, `SimulationModel`, `InterviewTranscript`). The `task` ontology describes activities associated with the creation

[2] http://www.policygrid.org/ResourceV4.owl

[3] http://www.policygrid.org/TaskV3.owl

of resources (e.g. SimulationDataAnalysis, SimulationParameterExploration). These ontologies together provide the underlying framework which defines the provenance for a piece of evidence. Moreover, our work on capturing *scientist's intent* provides additional information on how experiments were conducted, giving an improved insight into the evidence creation process.

This paper is organized as follows. Section 2 introduces our motivation through the use of a workflow example, section 3 presents an ontology for capturing scientist's intent. In section 4 we discuss the requirements for a semantic workflow infrastructure supporting our scientist's intent ontology. In section 5 we present some examples of how scientist's intent can enrich workflows. In section 6 we discuss issues arising from application of this approach to the domain of social simulation, and finally in section 7 and 8 we discuss related work and our conclusions.

2 Motivation and Example

Recent activities in the field of social simulation [14] indicate the need to improve the scientific rigour of agent-based modelling. One of the important aspects of any scientific activity 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 under which the experiment was carried out.

In a previous project, FEARLUS-G [15], we tried to meet the needs of agent based modelling using Semantic Grid technologies [2]. FEARLUS-G provided 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 experiments, hypotheses, parameters, simulation runs, and statistical procedures. We demonstrated 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 such experiments truly transparent.

We have identified a number of scenarios through interaction with collaborators from the social simulation community. We now present a simple example using a virus model developed in NetLogo[4]: 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 [16] 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.

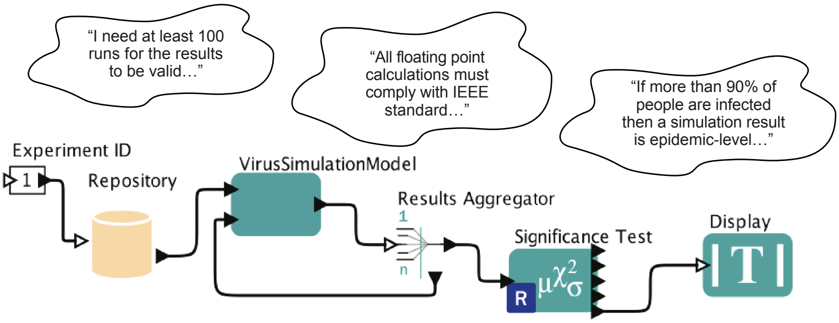
"I need at least 100 runs for the results to be valid…"
"All floating point calculations must comply with IEEE standard…"
"If more than 90% of people are infected then a simulation result is epidemic-level…"
Experiment ID
1
Repository
VirusSimulationModel
Results Aggregator
1
n
Significance Test
R
$\mu\chi^2_\sigma$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

Fig. 1. Example of Simulation Workflow

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

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

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 also be 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 scientist's 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).

3 Scientist's Intent

As part of our approach we have developed an ontology for capturing the scientist's intent based upon goals and constraints. Before discussing this ontology we need to specify some of the concepts and properties associated with workflows:

- **Workflow model** - The representation of the flow of data between tasks needed to complete a certain (in-silico) experiment.
- **Workflow activity** - A basic task in the workflow or a sub-workflow. Properties associated with a workflow activity are of two types: a) properties describing the activity itself (e.g. `Name`, `Type`, `Location`) b) properties describing the status of the activity at run-time.
- **Abstract workflow activity** - An abstract view of workflow activity that does not map to a specific task but its instantiation is decided at run-time.
- **Workflow links** - Indicate the temporal relationship between workflow activities e.g. the pipeline between workflow activities. This relationship is established by combining workflow activities' inputs and outputs. A typical property is the datatype of workflow inputs and outputs.

This leads us to the definition of goals and constraints associated with a workflow experiment:

- **Constraints** - A formal specification of a restriction on the properties of workflow activity (single task or sub-workflow), workflow activity at run-time, and workflow links (inputs and outputs).
- **Goal** - A formal specification of a desired state which is defined by a sub-set of workflow activity (single task or sub-workflow), workflow activity at run-time, and workflow links (inputs and outputs).

Based on the definitions of goal and constraint given above we propose an ontology for capturing scientist's intent associated with a workflow experiment, as shown in Figure 2. We begin by defining a WorkflowExperiment which represents a specific instance of a workflow model used to conduct a scientific experiment. A Workflow Experiment is designed to automate one or more tasks defined in the PolicyGrid task ontology (e.g. DataAnalysisTask, DataCollectionTask, etc.). A WorkflowExperiment contains one or more ComputationalResource instances which define the computational services (Grid, Web or local) associated with a workflow activity. Each ComputationalResource might have an associated ontology describing the resource as an entity but also describing properties of the resource at run-time. The metadata associated with ComputationalResource instances during the execution of the workflow provides information about the WorkflowState. A WorkflowExperiment has one or more instances of a WorkflowState capturing the temporal changes of workflow metadata. This is based on the idea of *Abstract State Spaces* [17] where a particular execution of services denotes a sequence of state transitions $\tau = (s_0, \ldots, s_m)$ [18]. A WorkflowExperiment is performed by a WorkflowEngine which characterizes a specific software implementation, e.g. Kepler[16]. Each WorkflowEngine implementation supports zero or more WorkflowActions, e.g. stop workflow, pause workflow, show message.

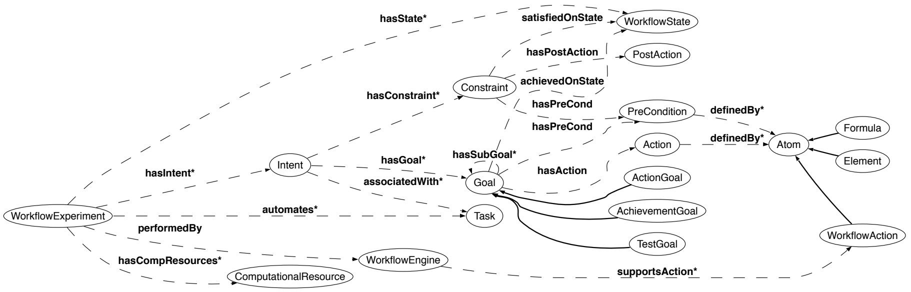

Fig. 2. Scientist's Intent Ontology

Central to our approach is the concept of Intent which is characterized by a set of Goal and Constraint statements. A WorkflowExperiment might have zero or more Intent instances. Although, from the definition above, Goal and Constraint are conceptually different, they share similar properties as they both have a PreCondition and a Action. Both properties are based on their constituent Atoms which can take the form of a metadata Element, a Formula or a WorkflowAction. As in SWRL[5] a condition is a conjunction of its Atoms. A PreCondition is a specific condition on the WorkflowState that can satisfy the Constraint or is achieved by the Goal [18]. An Action is an artifact that we use to trigger actions where the workflow engine is not able to reason itself about formal goals and constraints. Ideally a workflow engine would be fully aware of the goals and constraints defined in the *scientist's intent* and therefore be

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

able to reason about them but unfortunately this is not the case for most of the workflow engines currently available. In a *scientist's intent* aware workflow engine the planning and scheduling of the workflow execution can be optimized based on goals and constraints. Therefore, the concept of `PreCondition` is sufficient to represent both `Goal` and `Constraint` instances. However, most of the available workflow engines cannot be made fully compatible with scientist's intent without a major re-implementation, and therefore the concept of `Action` is required to overcome such limitations by providing additional metadata about the workflow state when goals are achieved and constraints are satisfied. Examples of such goals and constraints and their use will be presented later in this paper. However, to illustrate our approach, the scientific intent reflected in the example in Figure 1 can be represented as a combination of goals and constraints as follows:

- **Goal:** Run enough simulations to provide valid results to support the hypothesis. `(valid-run > 100)`
- **Constraint:** Significance Test has to run on a platform compatible with IEEE 754. `(platform = IEEE 754)`.

In our view details of the intent need to be 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 proposed here, a new experiment might be created just by changing the rules but not the underlying operational workflow.

4 Semantic Workflow Infrastructure

In this section we present a semantic workflow infrastructure solution based on the scientist's intent ontology described above, 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 manages goals and constraints of the experiment based on the workflow metadata.

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

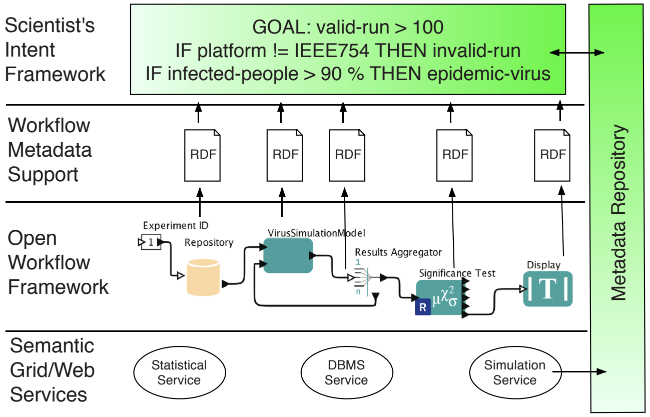

Fig. 3. Scientist's Intent Framework

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.

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`).

Central to our idea of capturing intent is the use of Semantic Grid services to perform the activities defined in the workflow. The main benefit of using such metadata enriched services is that they provide supporting information so that shared terms (e.g. `virus`, `experiment`, `simulation model`, `floating point standard`, etc.) can be used in the context of scientist's intent.

We have identified SWRL[6] (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 (`PreCondition`) and consequent (`Action`). This formalism is suitable for capturing scientist's intent, as the rules can capture the logic behind goals and constraints, while the ontology and metadata about the workflow provide the 'knowledge base' upon which the rules can operate. We selected the Bossam rule engine[7] for use within our architecture as it seamlessly integrates SWRL, OWL ontologies and RDF instances.

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

[7] http://bossam.wordpress.com/

Semantic grid services can provide different type of metadata: metadata about the service itself or metadata about the service execution at runtime. The latter involve many thousands of triples, and therefore a repository component is required to store such data. The Scientist's Intent Framework can then make use of the repository to extract metadata necessary to validate the rules but also to store any additional metadata (e.g. inferred statements).

5 Scientist's Intent & Workflow

We will now present some examples of goals and constraints to illustrate the benefits of scientist's intent in term of enriching workflow output, assisting workflow execution and provenance support.

5.1 Scientist's Intent to Assist Workflow Execution

As mentioned earlier, `Action` statements are used to add additional metadata to the workflow state if the workflow engine is unable to reason directly about goals and constraints. The example constraint below is used to check *if the significance test activity is running on a platform compatible with IEEE 754* as otherwise it will produce invalid results.

```
PreCondition:
  significanceTest( ?x1 ) ∧
  platform( ?x2, ‘‘IEEE754’’ ) ∧
  runsOnPlatform( ?x1,?x2 ) ∧
  hasResult( ?x1, ?x3)
Action:
  hasValidresult( ?x1, ?x3 )
```

If the significance test activity (see Figure 1) was defined as an abstract activity and the workflow engine was capable to interpret constraints directly, the selection of an appropriate significance test service could be made based on the pre-condition above.

The goal described below is used to specify the main goal for an experiment, i.e. to *run more that 100 valid simulation runs*. However we do not know a-priori how many simulation runs will be invalid.

```
PreCondition:
  significanceTest( ?x1 ) ∧
  hasResult( ?x1, ?x2 ) ∧
  hasValidresult( ?x1, ?x3 ) ∧
  more-than ( ?x3, 100)
```

The constraint below is used to check if the results of a particular simulation are invalid, specifically to determine *if the number of infected people (in a particular run) is greater than the number of not immune people (in the entire population)*.

```
PreCondition:
  population( ?x1 ) ∧
  virusMode( ?x2 ) ∧
  testPopulation( ?x2, ?x1 ) ∧
  hasModelRun( ?x2, ?x3 ) ∧
  notImmunePeople( ?x1, ?x4 ) ∧
  infectedPeople( ?x3, ?x5 ) ∧
  more-than( ?x5, ?x4 )
Action:
  hasInvalidRun( ?x2, ?x3) ∧
  stop(?x2) (Workflow Action)
```

Actions based on scientist's intent (e.g. `stop(?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 workflow tool to operate with our scientist's intent framework by registering the events that it is capable to detect and perform. These include: stop and pause the workflow, exit from a loop, show warning and error messages, prompt the user for information or intervention, display activity status.

5.2 Scientist's Intent to Enrich Workflow Output

Using the scientist's intent formalism it is possible to capture special kinds of constraints whose purpose is to enrich workflow outputs. While the previous goals and constraints support the verification and execution of the workflow by identifying invalid results or simulation runs, the constraints defined below aim 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*.

```
PreCondition:
  virus( ?x1 ) ∧
  virusModel( ?x2 ) ∧
  testVirus( ?x2, ?x1 ) ∧
  hasModelRun( ?x2, ?x3 ) ∧
  infectedPeople( ?x3, ?x4 ) ∧
  more-than ( ?x4, 90% )
Action:
  isEpidemic( ?x1 )
```

The new metadata resulting from the application of this constraint (`isEpidemic ( ?x1 )`) can be used as part of a *PreCondition* on other goals and constraints or as an annotation about the workflow outputs to facilitate analysis of the experimental results.

5.3 Provenance Support

As explained earlier in this paper, provenance is important for documenting the process that leads to a particular resource. We established that traditional provenance frameworks are not sufficient for all applications (e.g. policy appraisal) as it is very important to understand *why* particular steps in the process have been selected. We think that scientist's intent can be used to provide the important *why* context. For example, consider some of the constraint examples presented earlier. When looking back at the provenance of a simulation experiment it would be possible to determine *why* a particular statistical service had been selected (`platform (?x2,‘‘IEEE754’’)`) or *why* a particular simulation result was invalid (`notImmunePeople < infectedPeople`);

6 Case Study Discussion

We are exploring the use of workflow technologies in combination with our scientist's intent framework with a group of simulation modellers. Their work focuses on simulation of rural land use change in the Grampian region of Scotland over the past few decades, and on likely future responses to climate change, and to regulatory and market responses to it. The work is being supported by the Scottish Government through the research programme "Environment - land use and rural stewardship", and by the European Commission through the CAVES project[8]. It is in planning sequences of simulation runs, and associated statistical testing, required to validate, refine and use the models that it is planned to use workflow technologies.

Although a full evaluation of the scientist's intent framework in this real case-study environment has not yet been carried out, a number of issues about enabling agent-based models to work with our framework have been raised.

As mentioned earlier, one of the key issues for agent-based modelling has been the question of repeatability [19]. Authors reporting replication of agent-based modelling work have often commented that considerable interaction with the developers of the original model was necessary [20]. Using workflow technologies with the scientist's intent framework, it will be possible to record metadata about activities undertaken using a piece of modelling software and goals and constraints associated with it. This means that if one has access to the software from which conclusions were derived, it is possible to reconstruct the simulation output basis on which the conclusions were reached. This is also a timely contribution in the context of increasing demands from funding bodies for recognised standards to audit traceability of scientific results (in some cases, under the auspices of ISO9001).

A full replication of a piece of agent-based modelling work would ideally involve a reimplementation of the model, without any code reuse from the original software. Workflow metadata and scientist's intent is also useful here, as the re-implemented model can then presumably undergo the same processes used to derive conclusions as were used with the original model. However, deeper ontological support covering the structure of the model itself would facilitate the reimplementation process, and (related to this) provide a basis for verifying the similarity of the original and reimplemented models. Whilst some such information may be covered in accompanying documentation if available (often in the form of UML diagrams), ontological support can capture meanings in software representations not covered by the semantics of the implementing programming language [21], as well as providing a resource with which automated reasoning can be used.

[8] http://cfpm.org/caves/

7 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 [16]. However, the underlying representation of the workflow remains the same (data and control flow). For example the language BPEL [6], originally designed for business, 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. 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 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 [7] 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[9] 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 [5] is a language for building models as clustered graphs of entities with inputs and outputs. Like Taverna with XScufl, Kepler [16], is a workflow tool based on the MoML language where Web and Grid services, Globus Grid jobs, and GridFTP can be used as components in the workflow. Kepler extends the MoML language by using Directors which define execution models and monitor the execution of the 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. [22] 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 workflow templates and workflow instances. This description

[9] www.mygrid.org.uk

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.

Also relevant to our work is the model of provenance in autonomous systems presented by Miles et al. [23]. This model combines a description of goal-oriented aspects of agency with existing provenance frameworks in service-oriented architectures.

8 Discussion

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 scientist's 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. [4] 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. We have the underlying provenance framework to capture metadata about resources and tasks. The scientist's intent framework provides additional metadata about goals and constraints associated with a task (or set of tasks). Moreover through the use of `Action` the scientist's intent framework can also provide additional provenance generated from goals and constraints. However, we acknowledge that to truly understand the intent of the scientist a meta-level interpretation of all the above sources of provenance is necessary and this is beyond the current scope of our work.

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

1. Pennington, D.: 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)
2. Roure, D.D., Jennings, N., Shadbolt, N.: The semantic grid: a future e-science infrastructure. Grid Computing: Making the Global Infrastructure a Reality (2003)
3. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: Grid services for distributed system integration. Morgan Kaufmann, San Francisco (2002)
4. Hielkema, F., Edwards, P., Mellish, C., Farrington, J.: A flexible interface to community-driven metadata. In: Proceedings of the eSocial Science conference 2007, Ann Arbor, Michigan (2007)
5. Lee, A., Neuendorffer, S.: Moml — a modeling markup language in xml —version 0.4. Technical report, University of California at Berkeley (2000)
6. Andrews, T.: Business process execution for web services, version 1.1 (2003), `ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf`
7. Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M., Wipat, A., Li, P.: Taverna: a tool for the composition and enactment of bioinformatics workflows. Bioinformatics Journal 20(17), 3045–3054 (2004)
8. Groth, P., Jiang, S., Miles, S., Munroe, S., Tan, V., Tsasakou, S., Moreau, L.: An architecture for provenance systems. ECS, University of Southampton (2006)
9. Goble, C.: Position statement: Musings on provenance, workflow and (semantic web) annotation for bioinformatics. In: Workshop on Data Derivation and Provenance, Chicago (2002)
10. Zachman, J.A.: A framework for information systems architecture. IBM Syst. J. 26(3), 276–292 (1987)
11. Greenwood, M., Goble, C., Stevens, R., Zhao, J., Addis, M., Marvin, D., Moreau, L., Oinn, T.: Provenance of e-science experiments. In: Proceedings of the UK OST e-Science 2nd AHM (2003)
12. Chorley, A., Edwards, P., Preece, A., Farrington, J.: Tools for tracing evidence in social science. In: Proceedings of the Third International Conference on eSocial Science (2007)
13. Wilson, E.: An introduction to scientific research. McGraw-Hill, New York (1990)
14. Polhill, J., Pignotti, E., Gotts, N., Edwards, P., Preece, A.: 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 (2007)

15. Pignotti, E., Edwards, P., Preece, A., Polhill, G., Gotts, N.: Semantic support for computational land-use modelling. In: Proceedings of the Fifth IEEE International Symposium on Cluster Computing and Grid (CCGrid) 2005, vol. 2, pp. 840–847. IEEE Press, Los Alamitos (2005)
16. Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jeager, E., Jones, M., Lee, E., Tao, J.: Scientific workflow management and the kepler system. Concurrency and Computation: Practice and Experience (2005)
17. Keller, U., Lausen, H., Stollberg, M.: On the semantics of functional descriptions of web services. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 605–619. Springer, Heidelberg (2006)
18. Stollberg, M., Hepp, M.: A refined goal model for semantic web services. In: Proceedings of the Second International Conference ion Internet and Web Applications and Services. ICIW apps 2007, pp. 17–23 (2007)
19. Hales, D., Rouchier, J., Edmonds, B.: Model-to-model analysis. Journal of Artificial Societies and Social Simulation 6(4) (2003)
20. Axtell, R., Axelrod, R., Epstein, J., Cohen, M.D.: Aligning simulation models: A case study and results. Computational and Mathematical Organization Theory 1(2), 123–141 (1996)
21. Polhill, J.G., Gotts, N.M.: Evaluating a prototype self-description feature in an agent-based model of land use change. In: Amblard, F. (ed.) Proceedings of the Fourth Conference of the European Social Simulation Association, Toulouse, France, September 10-14, pp. 711–718 (2007)
22. Gil, Y., Ratnakar, V., Deelman, E., Spraragen, M., Kim, J.: Wings for pegasus: A semantic approach to creating very large scientific workflows. In: Proceedings of the 19th Annual Conference on Innovative Applications of Artificial Intelligence (IAAI), Vancouver, British Columbia, Canada (2006)
23. Miles, S., Munroe, S., Luck, M., Moreau, L.: Modelling the provenance of data in autonomous systems. In: AAMAS 2007: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems, pp. 1–8. ACM, New York (2007)