Putting Semantics in Grid Workflow Management: the OWL-WS approach

Stefano Beco[1], Barbara Cantalupo[1],
Nikolaos Matskanis[2], Mike Surridge[2]
[1]*DATAMAT S.p.A, Rome, Italy*
{stefano.beco, barbara.cantalupo}@datamat.it
[2]*IT Innovation, Southampton, UK*
{nm,ms}@it-innovation.soton.ac.uk

Abstract

Providing a complete and effective framework for describing and managing Web and Grid Services is becoming a key point in the Semantic research area. Also supporting process description and enactment, by means of composition of multiple resources, emerged as a fundamental requirement in the Web and Grid context. Still Semantic Workflow definition and management has not been addressed yet from a general and integrated perspective.

Within NextGRID[1] project, the need of a semantic workflow representation language to be used for specifying both application workflows and adaptive business processes (also expressed as workflow policies) emerged from the idea of adopting business processes as architectural components in a next generation Grid.

Language definition was based on the development of a Semantic Workflow and Service Model that allows us modelling concept like abstract and concrete services and workflows. Based on the analysis of existing technologies, OWL-S was selected as the most appropriate language to fit our model. Specification of extensions to support the proposed model resulted in the OWL-WS (OWL for Workflow and Services) ontology definition that should be considered as a first step towards an effective Semantic Workflow framework specification based on commonly agreed and recognized standard.

1. The role of Workflows and Semantics in the NextGRID context

NextGRID project [1] developed the idea of defining a workflow-centric model of Grid execution (a Grid Virtual Infrastructure Model) that is capable of handling different business processes (expressed as workflow policies) according to dynamic adaptive rules. This implies the need for a workflow representation language allowing defining dynamic workflows and specifying semantic information to be used in the evaluation and enactment process. Specification and analysis of a dynamic workflow enactment model enable to resolve abstract application workflow into concrete infrastructure ones, producing intermediate workflow with the application of policy workflow, resulted in the need of a representation model providing the ability to perform the following functionalities:

- *Abstract workflows definition*. It must be possible to describe workflows without specifying a binding of each task to a service, so the bindings can be added at run-time.
- *Semantic task description*. Each task in an abstract workflow should carry a description, or at least a task type, allowing a service providing that task to be found at run-time.
- *Workflow substitution*. It must be possible to define bindings of abstract tasks to more detailed workflows that can be inserted into the enactment at run-time.
- *Higher-order workflows management*. It must be possible to treat workflow continuations as data, so a task can take a workflow as input and return another workflow for execution.

[1]The work presented in this position paper has been supported by NextGRID project. NextGRID is a project funded by the European Commission's IST programme of the 6th Framework Programme (contract number 511563). See website at www.nextgrid.org.

2. Semantic Workflow Technology Overview

Workflow itself is quite a wide concept and technology whose meaning and usage can vary according to the different computational areas is applied on. Applying Semantic to Grid Workflows surely need inputs and suggestion from at least three different research areas as depicted in Figure 1.

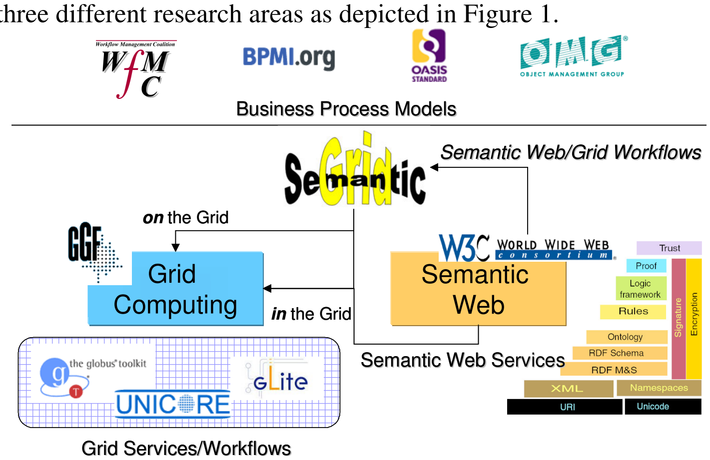

Figure 1: Semantic Workflows Technology Reference Overview

From the analysis of existing languages and models in these research areas [2] we understood that current technology either in Semantics or Grid or Business Processes environment is not yet mature enough to provide a standard approach to the semantic workflow management in Web Services and in Grid systems frameworks. Still we felt that it did not make sense to *define yet another workflow language.*

We considered semantic issues as the most challenging and selected OWL-S as the best candidate to accomplish our task because it provides the ability (currently missing in WSDL-S) to express control and data flows needed to model workflows. Moreover, it was developed on top of the standard semantic web technology and therefore is fully integrated in a standardization effort that is not yet the case for the WSMO proposal. It must be considered that new proposals in this area emerged since the time we started our work like for instance Semantic Web Service Framework. However, we think that the underlying Semantic Workflow and Service Model and its future development can be a valuable reference for semantic workflow management not depending on the specific ontology/language used to represent it.

3. Semantic Workflow and Service Model Foundations

In order to define a suitable Semantic Workflow and Service Model we formalized main aspects that characterize a workflow. We based our model on two simple assumptions:

- Service is the basic workflow component: based on the context of a Grid architecture based on a SOA model, Semantic Grid Service (Service) is a Grid service, specified according to the OGSA glossary, that is defined and managed using some kind of semantic information.
- Self-similarity: Service and Workflows are similar entities from different level perspective and therefore can be viewed as the same functional entity and therefore can be managed in the same way (see Figure 2).

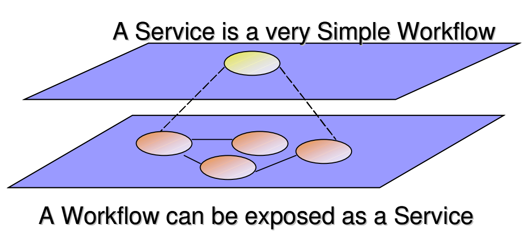

Figure 2: Self-similarity Representation

Concerning information that allows describing a workflows it can be divided in the following groups:

- ***Properties*** describing Service aspect from both functional (input, output, precondition, results) and non functional, that is semantic perspective (e.g. cost, QoS).
- ***Process*** describing relationship between different components (e.g. data flow, control flow, events).
- ***Execution*** providing information on the implementation of the service/workflow (e.g. service endpoint).

Workflow components can be therefore identified by their information:

- ***Service*** has properties and may have execution information (implementation)
- ***Workflow*** has properties, process information (flow information) and may have execution information (implementation)
- Services (and Workflows) can be described as ***Abstract or Concrete*** according to the kind of information that they currently have (see Figure 3).

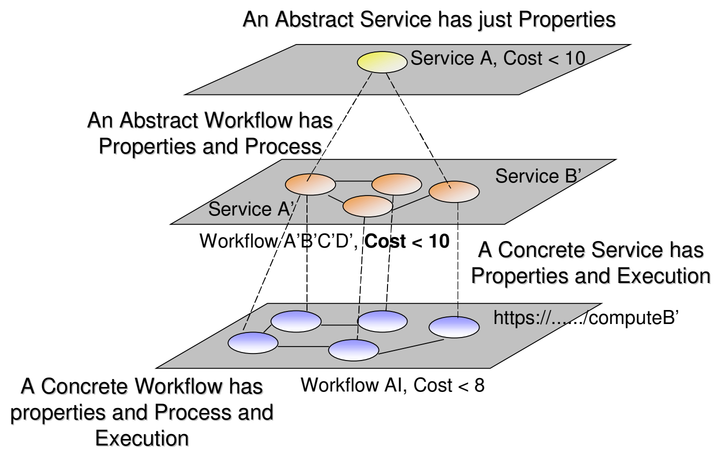

Figure 3: Semantic Workflow Components Representation

5. OWL-WS Language Model

OWL-WS is the semantic workflow language we defined to represent the previously introduced Semantic Workflow and Service Model. OWL-WS stands for "*OWL for Workflows and Services*" and is a workflow (and service) ontology fully based on OWL-S, with several changes/extensions. A more formal description is provided in [2].

Just as a reminder, OWL-S Upper Ontology is composed by the following elements:

- The class ***Service*** provides an organizational point of reference for a declared service.
- The ***Service Profile*** tells "what the service does" including description of what is accomplished by the service, limitations on service applicability and quality of service.
- The ***Service Model***, and more specifically ***Service Process*** subclass, tells "how to use the service", also detailing, where necessary, the step by step processes leading to the outcomes (Control/Data flows).
- The ***Service Grounding*** specifies the details of how the service can be accessed. Typically a grounding will specify a communication protocol, message formats, and other service-specific details.

The complete OWL-S model is adopted as it is in OWL-WS to represent Concrete Services and it is easily adapted to represent Concrete Workflows (see Figure 4).

- A **Concrete Service** is modelled in OWL-WS as a Service with its own Profile, a Process and a Grounding, whose element(s) refer to a single implementation Service.

- A **Concrete Workflow** is modelled in OWL-WS as a Service with its own Profile, a Process and a Grounding, whose element(s) refer to a multiple implementation Services.

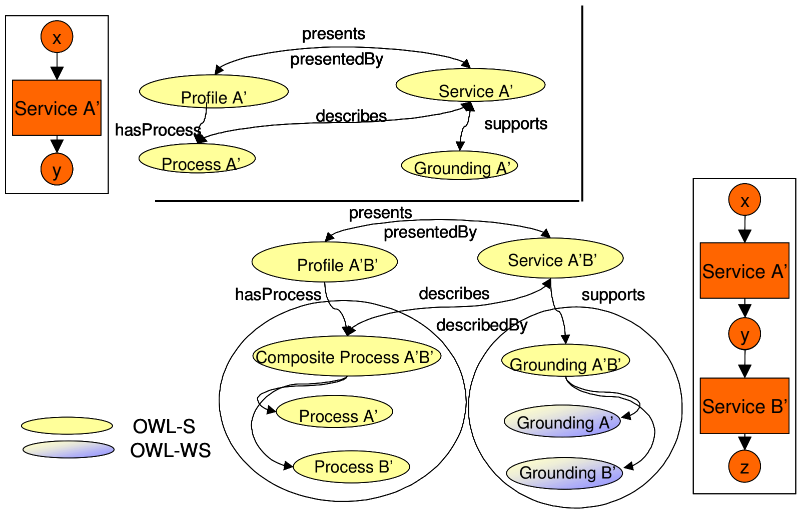

Figure 4: OWL-WS Concrete Components representation examples

OWL-WS uses the OWL-S concept of Composite Process for modelling workflows that are not only intra service, i.e. a sequence of calls to different operations of a single service, but also inter services processes.

Ability to represent Abstract Services and Workflows is provided defining an additional kind of Process:

- *OWL-WS Abstract Process is an Atomic Process having no link to any Grounding and provided with a new property "definedBy" that points to a Profile.*

Using this basic extension we can provide the following models (see Figure 5):

- An **Abstract Service** is modelled as a Service with its own Profile and Abstract Process. The Process property "definedBy" points to the Service Profile itself.
- An **Abstract Workflow** is modelled as a Service with its own Profile and Composite Process, whose elements are in turn Abstract Processes each one with its own Profile referred by the "definedBy" property.

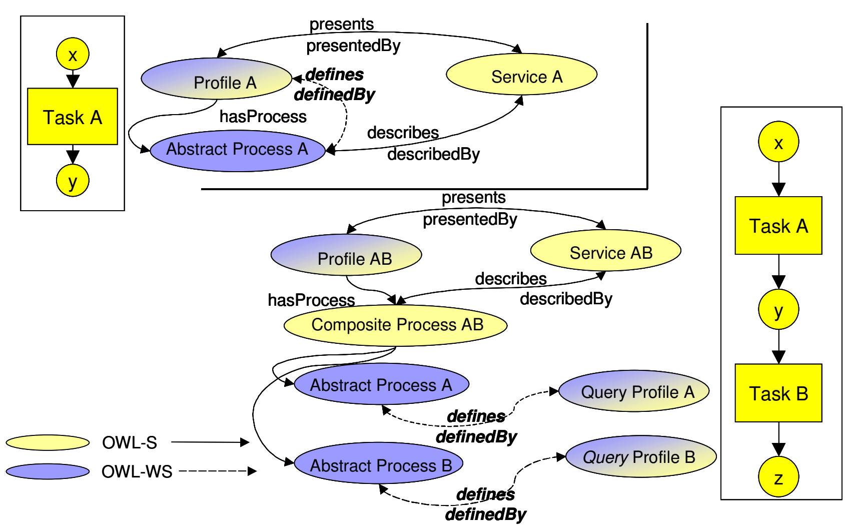

Figure 5: OWL-WS Abstract Components representation examples

The OWL-WS capability of modeling composition of OWL-S services (that means Service, Profile, Process components all together) allow specifying Profiles not only for the single services but also for workflow (and sub-workflows). This allows us to support ***Service Grouping*** management by providing information not only for the single Abstract Service but also for specific service groups modelled as inner workflows (see Figure 6).

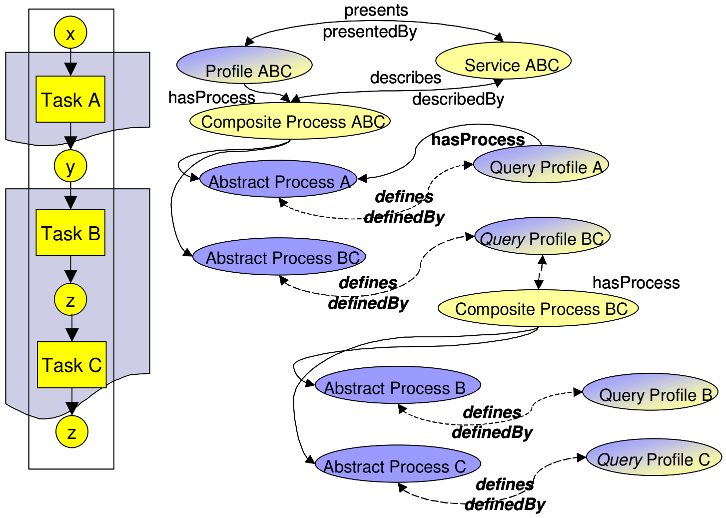

Figure 6: OWL-WS Service Grouping capability example

As a final note, it is worth noticing that in the OWL-WS language, Profile is the key element in Service and Workflow definition but it can be also described as any other data type. This provides OWL-WS with the ability to define and manage Higher-Order workflows.

6. Conclusions and Next Work

We developed a Semantic Workflow and Service model and related OWL-S extensions needed to support it, therefore defining an OWL-WS framework for managing adaptive Grid Service composition. Experiments will be soon performed specifying realistic business processes in order to really validate the model. Components of an integrated architectural framework are being developed comprising a user-friendly workflow programming tool supporting users in specifying and managing workflows, and the enactment engine (based on the Freefluo workflow enactor [3]), implementing the architectural integration between the workflow framework and the underlying Grid infrastructure.

However, to provide a comprehensive and effective Semantic Workflow and Service framework in the next generation Grid context a number of open issues still remain:

- Grid ontology adoption for specifying effective Profile properties.
- Semantic Web Rule Language investigation for specifying different kind of condition and managing inference in the workflow context.
- Model usability and feedback from different levels users.

Any comment and suggestion for a standard and effective specification of a Semantic Workflow model from the Semantic Grid community is absolutely welcome.

7. Main References

[1] Dave Snelling, Malcolm Atkinson, et al., *The NextGRID Conceptual Architecture V1.5*, January 17, 2005. It is available at NextGRID web site http://www.nextgrid.org.
[2] Stefano Beco, Barbara Cantalupo, Ludovico Giammarino, Mike Surridge and Nikolaos Matskanis. *OWL-WS: A Workflow Ontology for Dynamic Grid Service Composition*, accepted to the 1st IEEE International Conference on e-Science and Grid Computing, Dec. 5 - 8, 2005, Melbourne, Australia
[3] Tom Oinn, Matthew Addis, et al.. *Taverna: A tool for the composition and enactment of bioinformatics workflows,* Bioinformatics Journal 20(17) pp 3045-3054, 2004.