Problem-Solving Methods for Understanding Process Executions

Problem-solving methods are high-level, domain-independent, reusable knowledge templates that support the development of knowledge-intensive applications. The authors show how to use them to bolster subject-matter experts' understanding of process execution by implementing such methods into the Knowledge-Oriented Provenance Environment.

In the context of scientific data for computation-intensive disciplines such as physics, biology, and astronomy, provenance focuses on describing and understanding where and how data is produced, the actors involved in its production, and the processes applied before it arrived in the collection from which it's now accessed. In a typical discovery task, for example, scientists integrate data from various sources, filter the combined data according to some criteria, and then annotate it with information about the relationships they've just discovered. All the tasks applied in this process contribute to that data product's provenance record.

However, having all this information recorded together with the data product isn't enough—given the large amount of information, the provenance record requires an abstraction process before anyone can use it. Think of provenance information as a pyramid with four levels from the bottom up: data, organization, process, and knowledge.[1] Although most current provenance systems focus on the first three levels by providing means for recording and querying process documentation, other efforts approach the provenance problem from a semantic perspective in an attempt to tackle the knowledge level. These systems use domain ontologies in Semantic Web languages such as RDFS (www.w3.org/TR/rdf-schema) and OWL (www.w3.org/2004/OWL), which establish well-defined associations between the resources used during process documentation and the domain. This lets users build semantic provenance metamodels with the terminology necessary for meaningfully expressing provenance entities and the relationships between them.

But regardless of the approach taken for provenance gathering and representation, the documentation of a process's execution generates large quantities of heavily linked and annotated provenance data. As the size and complexity of processes increase, process documentation can become hard to assimilate and eventually unmanageable. Furthermore, the main beneficiaries of provenance information are subject-matter experts (SMEs) who don't necessarily have a strong background in computer science or, more specifically, provenance. An additional semantic layer with a higher level of abstraction could help address this gap.

As much as possible, our goal is to support

1521-9615/08/$25.00 © 2008 IEEE
Copublished by the IEEE CS and the AIP

Jose Manuel Gómez-Pérez
iSOCO
Oscar Corcho
Universidad Politécnica de Madrid and University of Manchester

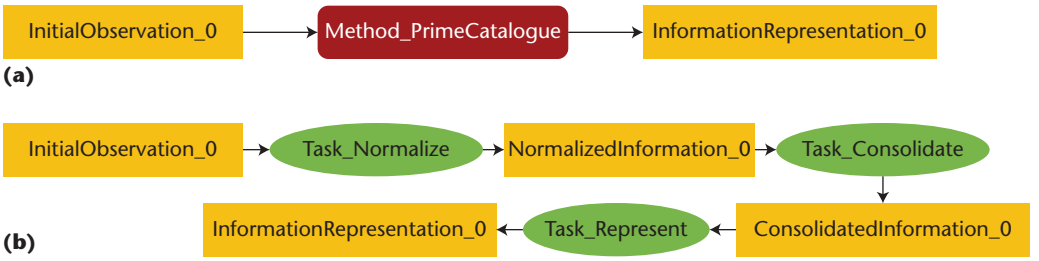

Figure 1. Interaction and knowledge-flow views. (a) The interaction view describes a problem-solving method (in red) in terms of its I/O roles (in yellow). (b) The knowledge-flow view shows a task decomposed into subtasks (green ellipses).

provenance interpretation by SMEs with little background in computer science. Thus, we use problem-solving methods (PSMs) as semantic overlays that represent provenance at multiple levels of abstraction. Our approach facilitates the user's understanding of how provenance information relates to process execution, simplifies process execution analysis by showing the overall process's decomposition into domain-level subprocesses, and offers visualizations of process execution at various levels of detail.

PSMs in Process Representation

PSMs originally emerged as reusable knowledge strategies that researchers could apply in different application domains to solve conceptually similar problems in terms of the goals to be achieved and the type of knowledge required. They're typically used to acquire knowledge and describe the primary rationale behind a process. But by applying PSMs to provenance analysis, we propose a novel way of using them: to interpret past events instead of model them. We can further exploit PSMs to facilitate provenance comprehension at our pyramid's knowledge level[2,3] by abstracting fine-grained provenance logs.

PSM frameworks such as the Unified Problem Solving Method Development Language[4] define four main types of knowledge resources: tasks, PSMs, domain models, and ontologies. Tasks provide high-level descriptions of the type of activity that we intend to accomplish by executing a particular process. Thus, we can view processes as occurrences (or instantiations) of tasks in a particular domain—for example, in the financial domain, a loan recommendation process is a particular occurrence of a generic assessment task,[5] whereas in the biological domain, a digestion process is an occurrence of task recombination.[6] Although tasks describe *what* a process's execution will achieve, PSMs describe *how* they'll do it: in short, PSMs define strategies, such as how to decompose tasks into simpler (sub)tasks, the steps required to accomplish each of them, and the knowledge to apply in each step. Finally, domain models describe the particular domain to which we apply tasks and PSMs, and ontologies provide the semantics required.

When focusing on PSMs as semantic overlays, we can enumerate their components[7] as follows:

- name,
- goal,
- subtasks,
- input tasks,
- output tasks,
- data flow between subtasks in terms of data roles,
- input roles,
- output roles,
- control flow over the subtasks, and
- suitability criteria.

We can also represent PSMs in three views:

- The *interaction view* (Figure 1a) describes a PSM (in red) in terms of its I/O roles (in yellow), providing a "black box" perspective.
- The *knowledge-flow view* (Figure 1b) shows how information is exchanged between subtasks (in green) as they consume and produce new knowledge to deal with the original task.
- The *decomposition view* (Figure 2) shows how PSMs decompose tasks into subtasks down to the level of primitive actions. As we continue the decomposition, the level of detail increases, producing more specific, fine-grained information about the PSM's strategies for a particular task. Figure 2, for example, shows that the prime catalogue method's strategy for accomplishing the catalogue task is to decompose it into three subtasks (normalize, consolidate, and represent), which interact with each other.

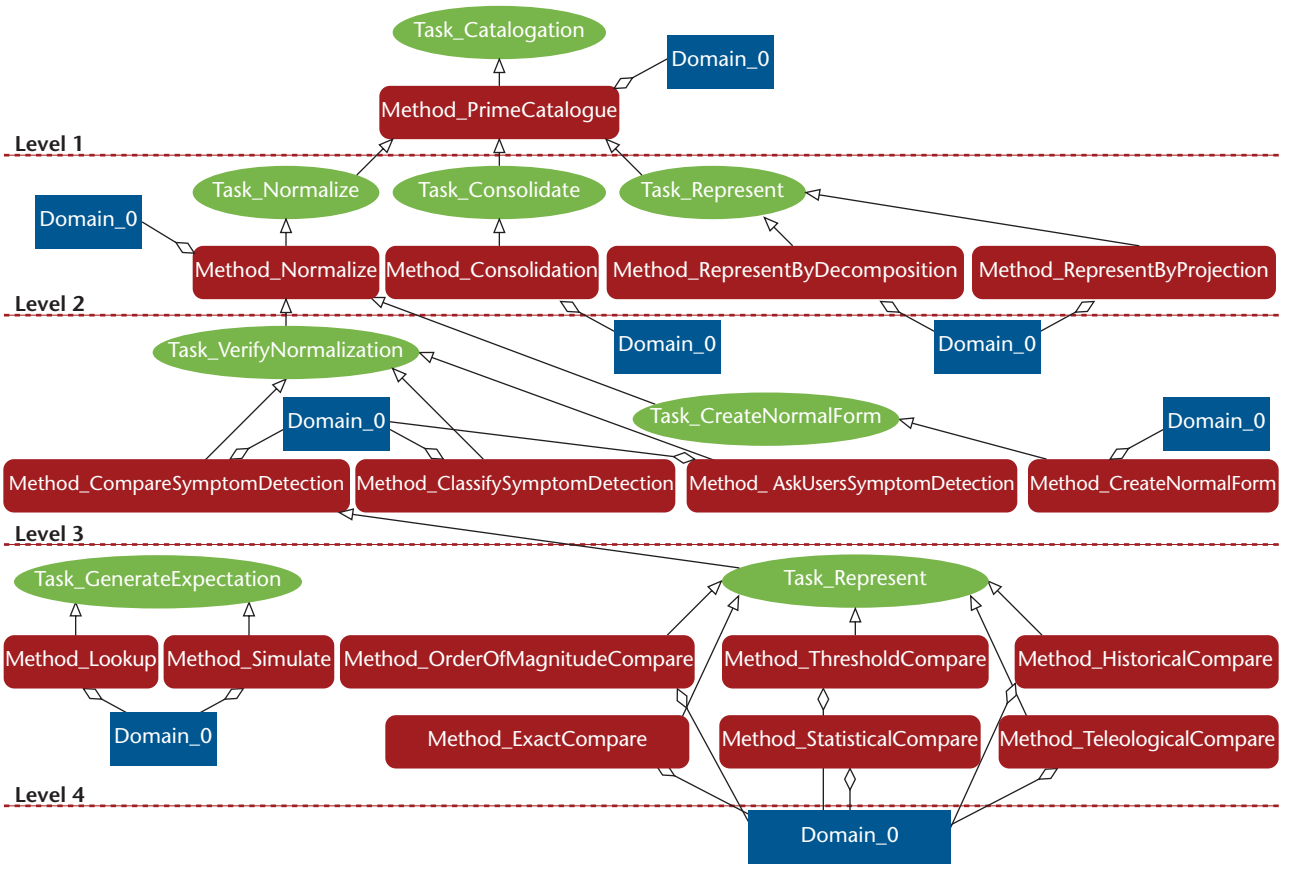

Figure 2. Decomposition view. The prime catalogue method's strategy for accomplishing the catalogue task is to decompose it into three subtasks (normalize, consolidate, and represent), but this view also offers alternatives, such as performing task compare with exact compare (*PSM exactCompare*) or threshold compare (*PSM thresholdCompare*).

The decomposition view also shows alternative methods, such as performing task comparison with exact compare (`PSM exactCompare`) or threshold compare (`PSM thresholdCompare`).

In the next section, we describe our approach's implementation in a real system.

A Knowledge-Oriented Provenance Environment

We developed the Knowledge-Oriented Provenance Environment (KOPE) as a stand-alone system the user can install to analyze provenance logs. KOPE requires the following knowledge resources to support user-oriented interpretations of provenance information at different levels of abstraction: a metamodel of PSM constructs and how they relate to each other, a library with a hierarchy of methods and instances of the PSM metamodel, and domain ontologies that describe the application domain.

As Figure 3 shows, KOPE's architecture has three building blocks: an underlying provenance infrastructure for documenting and querying process execution information, a PSM editor that lets users manage PSM libraries and domain ontologies as well as visualize provenance information at multiple levels of detail, and the KOPE engine, which uses the methods in the PSM libraries and ontologies to analyze process executions.

KOPE's underlying provenance infrastructure for process documentation and provenance querying is based on data structures identified by the Provenance-Aware Service-Oriented Architecture (PASOA) data model.[8] PASOA doesn't depend on a workflow enactor to produce process documentation—rather, it documents process execution as *p*-assertions. The PASOA model is flexible in terms of the contents of the information recorded in these *p*-assertions, which in turn lets us enrich process documentation with the semantic metadata automatically produced during process execution. Because domain and PSM entities are related by bridges, such metadata lets KOPE analyze provenance in terms of the domain,[9] ac-

cording to the generic process descriptions provided in the PSM library.

On the GUI side, KOPE uses an extended version of the Ontology Design Environment for Semantic Grid Services (ODESGS) PSM editor to invoke the KOPE engine for process execution analysis and visualization as well as to handle basic ontology management tasks.[10] ODESGS also supports mapping concepts from the domain ontology to PSMs' I/O roles via bridges. ODESGS can describe domain ontologies and methods in the PSM library independently to maximize their reusability across different domains.

As we described earlier, PSMs offer generic strategies for accomplishing the tasks of which domain-specific processes are occurrences. The KOPE engine's goal is to identify these tasks in the provenance store's process documentation. The KOPE engine detects task occurrences via twig join algorithms,[11,12] which, at each PSM decomposition level, allow matching the PASOA process documentation with the PSM's knowledge flow. In this process, the KOPE engine detects whether the twig between the execution's inputs and outputs occurs in the PSM's knowledge flow as well. The KOPE engine implements *twig_join*(D, $i(T)$, $o(T)$) as a Boolean function that checks whether a twig exists that joins $i(T)$ and $o(T)$ in D, where $i(T)$ is the set of input roles of T, $o(T)$ is the set of output roles of T, and D is the *p-DAG* of the documented process, returned by a provenance query.

The Provenance Challenge

In 2007, we evaluated KOPE in the context of the "Provenance Challenge" (http://twiki.ipaw.info/bin/view/Challenge/SecondProvenanceChallenge), a community-driven effort toward provenance standardization stemming from discussions first held at the International Provenance and Annotation Workshop (IPAW 06; www.ipaw.info). This initiative's goal is a comprehensive standard that will eventually ensure interoperability among different systems using compliant data models.

To date, the challenge has occurred twice: the first (2006) provided valuable insight into the various provenance approaches already existing in the community, whereas the second (2007) proposed a systematic approach for comparing different systems and representations of provenance data. Interoperability among provenance systems was the issue in the first challenge, thus the second challenge intended to understand the extent to which data in one model is translatable to or has no parallel in another model and how to trace data provenance across multiple systems, thereby adding value to all those systems.

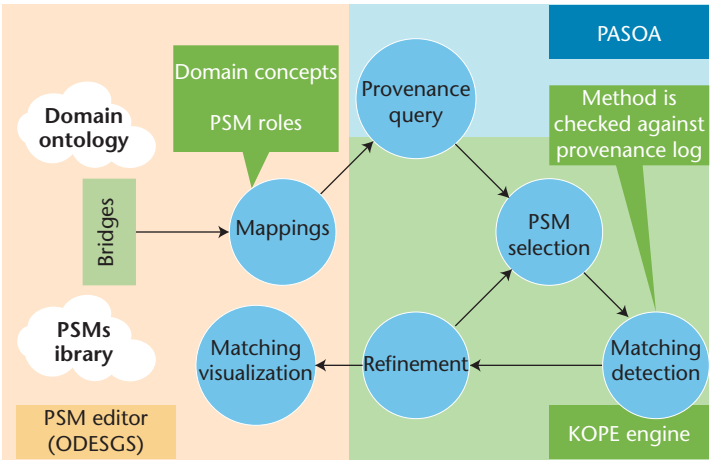

Figure 3. Knowledge-Oriented Provenance Environment (KOPE) architecture. The PSM editor (ODESGS) uses bridges to relate domain ontologies with PSM libraries. PASOA is the underlying low-level provenance infrastructure used to document process executions; the KOPE engine selects adequate PSMs from the library and abstracts the detailed provenance logs by matching them with the tasks into which they're divided.

Evaluation Setup

During both challenges, several worldwide teams evaluated the systems in the context of a workflow for creating population-based brain atlases from the fMRI Data Center's archive of high-resolution anatomical data. The workflow associated with this process comprises procedures and the data items flowing between them.

KOPE participated in the second challenge with a twofold goal: to evaluate its interoperation with other provenance systems, in particular with PASOA (because its infrastructure and data model support process documentation in the KOPE architecture), and to evaluate its capabilities for interpreting provenance information at the knowledge level. The knowledge resources KOPE used to analyze the brain atlas workflow's execution included the catalogue PSM library (specifically, the prime catalogue) and a domain ontology for brain atlases. The former describes strategies to solve the task of creating a catalogue in a given domain (in this case, the brain atlas), whereas the latter provides a description of brain images.

The prime catalogue method describes the task of creating catalogues at four different abstraction levels. The most abstract one, Level 1, is the method's top level, in which I/O roles are *initial obser-*

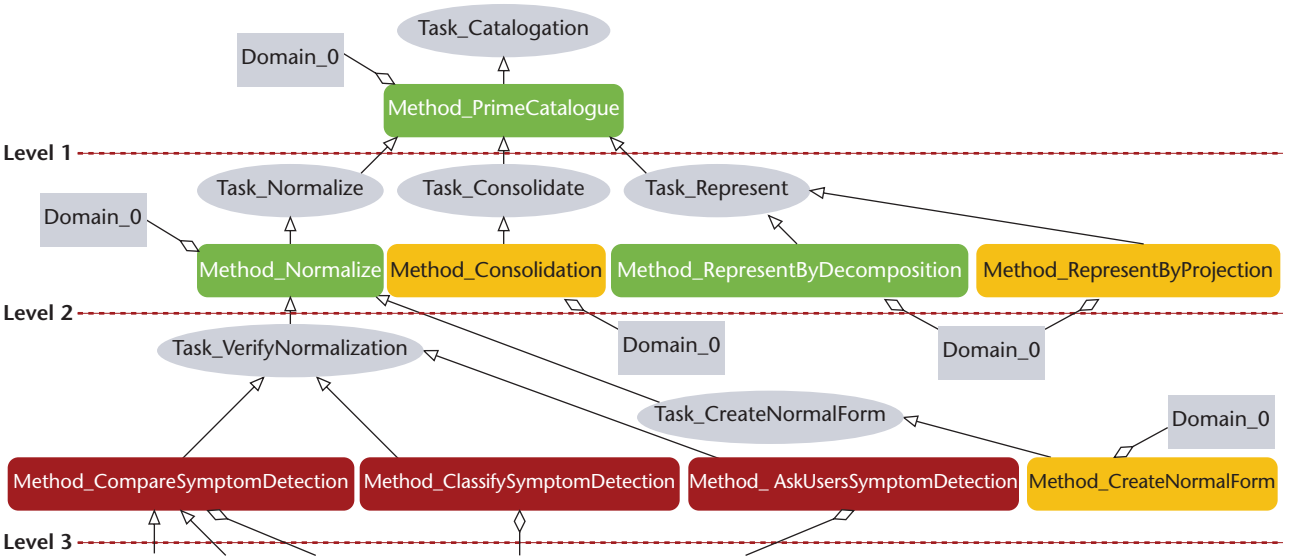

Figure 4. Analysis of the brain atlas creation process. Comparing the low-level provenance information with the prime catalogue method, green stands for a perfect match, yellow represents an imperfect match, and red indicates no match at all.

Table 1. Precision and recall per abstraction level of the prime catalogue method.

Abstraction level	Precision (%)	Recall (%)
1	100	100
2	83.3	100
3	25	100
4	0	0

vation and *information representation*, respectively. Being so abstract, this first level doesn't provide much information about the analyzed process; instead, Level 2 provides a first decomposition of the original task into finer-grained subtasks that describe the process in more detail. As mentioned earlier, these three subtasks are normalize, consolidate, and represent, and KOPE analyzes each of them via PSMs. The Level 2 task also shows two alternative methods—*represent by decomposition* and *represent by projection*—that contain different strategies for representing (or displaying) an item, which, in this case, is the result of the tasks preceding *represent* at this level (*normalize* and *consolidate*) in the PSM's knowledge flow view.

Task normalize reuses part of the PSM library for diagnosis and is the most complex of the hierarchy. A method of the same name decomposes this task into two subtasks (*verify normalization* and *create normal form*), which, together with the tasks that couldn't be further decomposed in Level 2 (*consolidate* and *represent*), form Level 3.

Evaluation Results

Because this evaluation's objective was to measure the accuracy of using PSMs as a semantic overlay for searching, recovering, analyzing, and eventually interpreting information about process execution from provenance data, we explain the results obtained in terms of precision and recall. Here, precision is the ratio of expected matches to actual matches, taking into account the different levels of decomposition the PSM library provided, whereas recall is the fraction of relevant matches the system returned.

We want to analyze the execution of the population-based brain atlas creation process, focusing on its validation with respect to the high-level process specification provided by the prime catalogue method. In this context, high precision and recall would indicate that this process's execution is perfectly compliant with the PSM-provided specification. However, high precision and low recall would also show that the SME didn't succeed in defining the correct mapping between domain entities and PSM roles via the required bridges, whereas low precision and high recall indicate that the PSM didn't provide a detailed enough specification of the process.

Figure 4 uses a color code to show the actual matching of the process documentation for brain atlas creation with the PSM library catalogue for each level of refinement that the prime catalogue method's decomposition view provided.

We originally designed the prime catalogue

method as a generic strategy for creating a catalogue of items in a given domain, but there are many ways in which to do so. Nevertheless, the prime catalogue method performed well during the challenge. With it, KOPE described the brain atlas creation process up to the method's third level of refinement, out of a maximum of four abstraction levels.

Overall precision and recall figures reached 75 and 100 percent, respectively; Table 1 shows the results at each abstraction level. We can therefore assert that the brain atlas creation process is compliant with the prime catalogue method's process specification and can be interpreted as an occurrence of a catalogue task (Level 1) that decomposes into three main tasks (Level 2) that occur sequentially. The last two tasks are completely compliant with the information recorded in the process documentation, but the first one (normalize) is only partially achieved because only one of its subtasks (*create normal form*, at Level 3) is reflected in the actual process. Low precision and high recall at Level 3 show that the prime catalogue method's process specification isn't so detailed at this level of decomposition. Table 1 shows this, where precision drops from 83.3 percent at Level 2 to 25 percent at Level 3; the 100 percent recall at Level 3 becomes 0 percent at Level 4.

With KOPE's participation in the second challenge, we've demonstrated our approach to provenance understanding has good results in terms of precision and recall when the level of detail that we're interested in is broad. However, when we move into the exact details about how each subprocess is executed, we need to improve our approach with better matching algorithms. This is part of our future work, in which we'll enhance some bottom-up approaches to provenance interpretation.[13] Our belief is that this hierarchical approach to provenance understanding is what SMEs require to understand which high-level processes are involved in a complex process execution.

As part of our future work, we'll also apply our approach to disciplines that don't just deal with scientific domains but that still have complex networks of processes, such as business process management in telecommunications, in which distributed groups of SMEs make decisions.

Acknowledgments

This work has been funded as part of the IST-2004-511513 EU project OntoGrid.

References

1. J. Zhao et al., "Using Semantic Web Technologies for Representing e-Science Provenance," *Proc 3rd Int'l Semantic Web Conf.* (ISWC 2004), Springer, 2004, pp. 92–106.
2. J. McDermott, "Preliminary Steps Towards a Taxonomy of Problem-Solving Methods," *Automating Knowledge Acquisition for Expert Systems*, Kluwer, 1993, pp. 225–255.
3. A. Newell, "The Knowledge Level," *Artificial Intelligence*, vol. 18, no. 1, 1982, pp. 87–127.
4. D. Fensel, "UPML: A Framework for Knowledge System Reuse," *Proc 16th Int'l Joint Conf. Artificial Intelligence* (IJCAI-1999), Morgan Kaufmann, 1999, pp. 16–23.
5. A. Schreiber et al., *Knowledge Engineering and Management: The CommonKADS Methodology*, MIT Press, 2000.
6. J.M. Gómez-Pérez, M. Erdmann, and M. Greaves, "Applying Problem Solving Methods for Process Knowledge Acquisition, Representation, and Reasoning," *Proc. 4th Int'l Conf. Knowledge Capture* (KCAP), ACM Press, 2007, pp. 15–22.
7. R. Benjamins, "Problem Solving Methods for Diagnosis and Their Role in Knowledge Acquisition," *Int'l J. Expert Systems: Research and Application*, vol. 8, no. 2, 1995, pp. 93–120.
8. S. Munroe et al., *Data Model for Process Documentation*, tech. report, School of Electronics and Computer Science, Univ. of Southampton, 2006.
9. S.C. Wong et al., "Provenance-Based Validation of E-Science Experiments," *Proc. 4th Int'l Semantic Web Conf.* (ISWC), LNCS 3729, Springer, 2005, pp. 801–815.
10. C. Goble et al., "ODESGS Framework, Knowledge-Based Annotation and Design of Grid Services," *Proc. 3rd Int'l Conf. Service Oriented Computing* (ICSOC 05), Springer, 2005, pp. 341–352.
11. N. Bruno, N. Koudas, and D. Srivastava, "Holistic Twig Joins: Optimal XML Pattern Matching," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, ACM Press, 2002, pp. 310–321.
12. L. Chen, A. Gupta, and M. Kurul, "Efficient Algorithms for Pattern Matching on Directed Acyclic Graphs," *Proc. 21st Int'l Conf. Data Eng.* (ICDE 05), IEEE CS Press, 2005, pp. 384–385.
13. Y. Simmhan, B. Plale, and D. Gannon, "A Framework for Collecting Provenance in Data-Centric Scientific Workflows," *Proc. Int'l Conf. Web Service* (ICWS 06), IEEE CS Press, 2006, pp. 427–436.

Jose Manuel Gómez-Pérez *is research manager at Intelligent Software Components (iSOCO) in Madrid. His research interests include Semantic Web technologies, distributed systems, and process knowledge. Gómez-Pérez is currently finishing his PhD in artificial intelligence at the Universidad Politécnica de Madrid. Contact him at jmgomez@isoco.com.*

Oscar Corcho *is an associate professor in the Ontology Engineering Group at the Universidad Politécnica de Madrid. His research activities include the Semantic Grid, the Semantic Web, and ontological engineering. Corcho has a PhD in artificial intelligence from the Universidad Politécnica de Madrid. Contact him at ocorcho@fi.upm.es.*