**Editor: Steffen Staab**
University of Koblenz-Landau
staab@uni-koblenz.de

# What Makes You Think That? The Semantic Web's Proof Layer

**Sergej Sizov,** *University of Koblenz-Landau*

A key motivation for Semantic Web technology is to better support self-organizing knowledge exchange between users. The idea of a human being browsing the Web should soon become old-fashioned. Personal work environments should be able to browse the Web automatically and find what the user is looking for, given what they know about the user's needs and the descriptive knowledge available on relevant remote sites. Yet what happens if the results seem strange, incorrect, or incomplete? I think I deserve an explanation—but can I really ask the computer, "*What* makes you think that?"

Nowadays, our expectations for receiving plausible reasons for the presented result are rather gloomy. The Semantic Web's decentralized nature makes finding the "what" difficult. Typically, we have no direct control over knowledge sources (such as journal articles), means of knowledge acquisition (such as workflows and knowledge extraction algorithms), and agents (such as expert users who verify extracted knowledge semi-automatically). By itself, this is a beneficial feature of the Semantic Web. The drawback is that the vast quantity of descriptive information about these aspects isn't available at the right time in the right place—namely, in the answer the user receives.

So, what is actually our "what"? There are several kinds of "what" information. For example, *data-what* describes the information and knowledge sources (such as which document was used for information extraction). *Transformation-what* describes how the system manipulates objects or data (such as which filtering algorithms it applied). *Personalization-what* describes the human influence on particular decisions (such as an expert's decision to include facts with low extraction confidence in the knowledge base). Finally, *infrastructure-what* describes the environment (such as parametrization of the natural language processing algorithm used, stop-word lists, and lemmatization settings) at knowledge acquisition.

Conceptually, this information contributes to the Semantic Web layer cake's *proof layer* (see figure 1). This layer is intended to explain the answers you get from the automated agents that consume the provided information. In other words, the proof layer is basically a layer of *what*. More concisely, this is the layer of *provenance knowledge*. In this column, I explore how we should represent, propagate, and integrate provenance knowledge.

## The proof layer's importance

Up to now, research has largely neglected the proof layer's specification and intended functionality, with the exception of some preliminary work.[1–3] However, it's fundamental to many Semantic Web applications and scenarios to understand how a result came about. Its practical role grows rapidly with the first successful prototypes of collaborative Semantic Web technology (for example, personal semantic wiki pages), which often require a deeper explanation of where the presented knowledge comes from and how the resulting conclusions have been constructed. It's not surprising that in past decades the Semantic Web community focused more on various aspects of tracking, representing, and exploiting specialized provenance aspects. Moreover, highly similar problem definitions have been recognized for Web 2.0 applications, distributed service-oriented architectures, desktop grids, peer-to-peer infrastructures, and almost all other collaborative environments.

## Semantic Web scenario

These problems integrate different views onto provenance and its collaborative management. To illustrate the interplay of different provenance problems, consider for example social network analysis for the computer science research community. In particular, exploring the relationships between authors and conference organizers is helpful for identifying interesting research partners, constructing personal research profiles, and forming program committees for upcoming conferences and workshops.[4]

Of course, in the presence of centralized authoritative bibliographic sources (such as DBLP or CiteSeer), constructing and mining the coauthorship or citation graph is straightforward and well understood. However, much of the valuable information about collaboration comes from "gray" publications (such as technical reports and proj-



1541-1672/07/$25.00 © 2007 IEEE
*Published by the IEEE Computer Society*

ect deliverables), co-organization of events such as summer schools and workshops, or cross-references between personal Web pages. This information is typically spread across research departments' HTML pages and can't be automatically explored. The workshop chair must manually browse hundreds of Web pages to learn about potential program committee candidates—time consuming but unavoidable.

The Semantic Web has great potential for supporting such scenarios by automatically acquiring and interpreting the desired information. By analyzing the Semantic Web pages of corresponding universities and departments, the system could identify facts about researchers' collaboration and past submissions to relevant conferences in minutes and could merge them into a ranked list of recommended candidates. This vision of solving our sample "program committee problem" clearly goes far beyond existing workarounds for HTML contents such as named-entity recognition or thematically focused (topical) crawling.

The proof of the results acquired from a specialized information system is (more or less) feasible. By analyzing received inputs and backtracking query-execution steps, I can usually identify facts that contributed to the particular recommendation. For these facts, one clearly defines common dimensions such as the provider (for example, DBLP, CiteSeer, or Google Scholar) or agent (for example, the portal administrator). In addition, one might hope to receive further portal-specific provenance hints (such as the knowledge extraction source—say, an ACM or IEEE publication database). With this solid proof in mind, one usually sets a "flat rate of trust" for sources such as DBLP or CiteSeer without worrying further about result provenance.

The situation could change in a general setting of the decentralized Semantic Web scenario. Sometimes, retrieved facts about coauthorship or collaboration between scientists might appear confusing, unusual, or surprising. For example, consider a strong record of collaboration between the International Semantic Web Conference (ISWC) best paper award winner and an ACM fellow in numerical algorithms. These facts are potentially the most interesting, indicating hubs between different research communities and paradigms—a recommendation for a good interdisciplinary program committee or ongoing project consortium. Or is it just a mix-up in the buggy knowledge extraction algorithm? A closer look at the result provenance should explain which facts and sources have contributed to the presented recommendation.

## What is provenance?

According to the *Oxford English Dictionary*, provenance can mean

- the fact of coming from some particular source or quarter; origin, derivation.
- the history or pedigree of a work of art, manuscript, rare book, etc.; concr., a record of the ultimate derivation and passage of an item through its various owners.

However, the intentional semantics of "provenance" has been frequently adjusted in the electronic-collaborative-applications domain. Here, the notion has had several custom interpretations, depending on the target system's execution environment, representation mechanisms, query language, and application-specific properties. In particular, there are three different (but important) viewpoints: Semantic Web provenance, database provenance, and workflow provenance.

### Semantic Web provenance

Recent work on Semantic Web provenance includes representational mechanisms for metaknowledge (that is, knowledge about knowledge, including provenance) and its use. For example, Jeremy Carroll and his colleagues proposed an application of named RDF graphs for publishing information on the Semantic Web.[5] This scenario implies that humans and agents have two basic roles: information providers and information consumers. Information providers publish information together with meta-information. Additionally, they might publish background information about themselves, such as their role in the application area. They might also digitally sign the published information. Information providers have different knowledge levels, intentions, and world views. So, from an information consumer's perspective, published graphs are the information providers' provenance claims. An information consumer can accept some of these claims and reject others. These choices are represented by the information consumer accepting named graphs. Different kinds of meta-information can be put into different named graphs. Obviously, different consumer tasks require sources with different provenance. So, information consumers will use different policies to decide which graphs they should accept and use in the specific application. These policies depend on the application area, the information consumer's subjective preferences and past experiences, and the relevant provenance information available. For instance, Li Ding and his colleagues described the use of provenance and trust information in the context of homeland-security analysis.[6] They addressed the description of an inference framework addressing these issues:

- capturing provenance information (they addressed this issue specifying general principles),
- using provenance to evaluate a semantic association's trustworthiness, and
- using provenance knowledge to prune search on the Semantic Web.

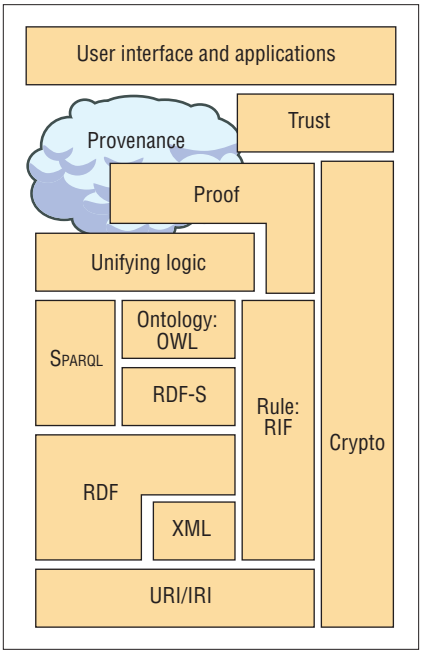


**Figure 1. The Semantic Web layer cake.**

### Database provenance

Database systems usually consider provenance as describing the data's origins and the process by which it arrived as a query answer. The established terminology distinguishes between where-provenance, why-provenance, and how-provenance:[6–8]

- *Where* is where the given fact or state-

```
@prefix rdf:  http://www.w3.org/1999/02/22-rdf-syntax-ns
@baseURI http://www.problem.com/sampleDocument

:G1 {                                              t1
        Bob collaboratesWith Mary.                 t2
        Mary worksInProject ProvenanceChallenge.
    }

:G2 {                                              t3
        Hugo collaboratesWith John.                t4
        John publishedOn ISWC.                     t5
        Mary publishedOn ISWC
    }
```

**Figure 2. Vision scenario: Samples of extracted knowledge.**

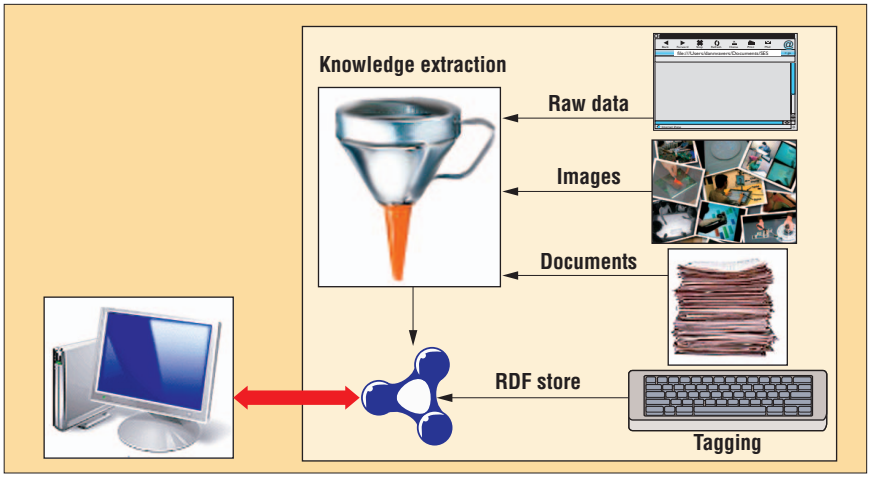


**Figure 3. A general application scenario.**

ment is physically serialized in one or more RDF statements (that is, "where does a given piece of data come from?").
- *Why* is the collection of facts or statements that contributed to produce the query answer, such as a composed statement ("which facts contributed to this answer?").
- *How* is how the query result was produced ("how did facts contribute to the answer?").

Researchers have studied the why/where/how provenance approach as an extension for relational databases (that is, data management systems based on relational algebra) and probabilistic databases,[9] and they later adopted it for RDF knowledge bases (that is, for semantics of the SPARQL query language). By defining custom (possibly different) interpretations for algebraic operations of Boolean formulas (built on tuple IDs as Boolean variables) for particular factors of provenance (such as agent, time stamp, source, and certainty dimensions), one can obtain the $m$-dimensional provenance record for the query result.[10]

The database community usually treats provenance as an extension of the core data model, coined *annotated relations*. Annotated relations are relations of a database where each tuple is annotated with an element of an arbitrary set $K$. The desired instantiation of $K$ is the set of Boolean formulas built from tuple identifiers and a distinct element 0. (NULL); The tuple identifiers (labels) are regarded as Boolean variables. Indeed, a Boolean expression of the result set built on tuple identifiers carries information about not only *which* triples have contributed to a variable assignment (why-provenance) but also *how* they contributed (how-provenance).

In my sample application scenario, I can assume that the search for program committee candidates with relevant collaboration profiles uses knowledge that was initially collected from Computer Science departments' Semantic Web pages and stored as RDF triples in the workshop chair's personal "active space,"[11] backed by a local RDF repository (see figure 2).

Figure 3 shows (in a slightly oversimplified form) the relevant facts that the departments of different universities might have obtained. The first graph in the example http://www.example.org/sampleDocument#G1 contains two statements automatically extracted from a research report (such as an MS Win Word document). In the local RDF repository, these statements have (internal) tuple IDs $t_1$ and $t_2$. The second graph with tuples $t_3$, $t_4$, $t_5$ contains facts a PDF parsing tool automatically extracted from a conference survey paper. For simplicity, figure 2 directly shows corresponding tuple IDs together with tuples of named graphs $G1$ and $G2$; in practice, you can realize annotated tuples via reification[12] or additional store-specific internal bookkeeping.

The extracted knowledge comes from different sources, at different time points, with different degrees of confidence in the extraction algorithm. The same RDF repository also captured this information. Figure 4 shows the metadata associated with extracted knowledge using the notion of Named RDF Graphs.[12,13]

Figure 5 shows the corresponding SPARQL-style query. Its syntax (and semantics) slightly deviate from the common notion of SPARQL. The clause WITH PROVENANCE forces the query processor to additionally construct the generated result set's proof.

Figure 6 shows the desired outcome (expressed using the Named Graphs syntax[12,13]) that the query processor could generate according to provenance management rules.[10] It consists of two parts. The first part (graph $G11$) contains facts that the extended SPARQL query processor constructed according to the query specification. The second part (graph $G11Meta$) provides the record on how-provenance that explains which tuples from the repository contributed to the outcome and how the result was constructed. The symbolic how-provenance record $(t_1 \wedge t_3) \vee (t_5 \wedge t_4)$ explains that the result set's tuples were generated by combining source records $t_1$ and $t_3$ (from named graphs $G1$ and $G2$) or source records $t_4$ and $t_5$ (from named graph $G2$).

By replacing custom interpretations of the operators $\wedge$ and $\vee$ for particular domains, $dom(A_i)$, I obtain corresponding interpretations for metaknowledge dimensions. I can easily redefine these definitions to better capture a specific provenance domain's particular behavior. For example, by redefining $certainty^{*}(A \wedge B) = \min(certainty(A), certainty(B))$, I change the probabilistic interpretation of certainty values to fuzzy interpretation:

$$
\begin{aligned}
&source(A \wedge B) = source(A) \cup source(B)\\
&source(A \vee B) = source(A) \cup source(B)\\
&agent(A \wedge B) = agent(A) \cup agent(B)\\
&agent(A \vee B) = agent(A) \cup agent(B)\\
&extractor(A \wedge B)\\
&\quad = extractor(A) \cup extractor(B)\\
&extractor(A \vee B)\\
&\quad = extractor(A) \cup extractor(B)\\
&certainty(A \wedge B) \qquad (1)\\
&\quad = certainty(A) \times certainty(B)\\
&certainty(A \vee B) = certainty(A) +\\
&\quad certainty(B) - certainty(A \wedge B)\\
&timestamp(A \wedge B)\\
&\quad = \min(timestamp(A), timestamp(B))\\
&timestamp(A \vee B)\\
&\quad = \min(timestamp(A), timestamp(B))
\end{aligned}
$$

By inspecting received answers, the expert user might realize that, for example, the recommendation of Hugo (an undergraduate student in John's group) is potentially inappropriate. To identify the mistake's source, the expert user looks more closely at the source document report01.doc. The user realizes that Hugo, a student assistant, formatted the research report. His name appears in document metadata fields that the extraction algorithm inappropriately interpreted as an indication of coauthorship.

### Workflow provenance

The Grid and Semantic Web Services communities tackle the provenance problem differently. They compose complex services to solve a given problem, typically via a workflow that specifies their composition. Interactions with services take place using messages constructed in accordance with service-interface specifications. Clients typically invoke services, which might themselves act as clients for other services (that is, *actor* can denote either a client or a service). A workflow's execution is called a *process*. Finally, the provenance of knowledge is defined as a formal specification of the process that led to the given result. At some abstraction level, provenance captures a notion of a causal graph, explaining how a data product or event came to be produced in an execution. So, by having a description of the process that resulted in a data item, the Grid and Semantic Web communities can explain how such a data item was obtained. The rich body of related service-oriented architecture work[14–16] has resulted in provenance-aware formal models and architectures.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns>
@baseURI http://www.problem.com/sampleDocument

:G3
  { :G1 source <http://example/researchReport.doc> .
    :G1 agent Hugo.
    :G1 extractor wordAnalyzer.
    :G1 certainty "0.9".
    :G1 timestamp "5/5/2007"
  }

:G4
  {
    :G2 source <http://example/provenanceSurvey.pdf> .
    :G2 agent Betty.
    :G2 extractor pdfAnalyzer.
    :G2 certainty "0.6" .
    :G2 timestamp "6/6/2006"
  }
```

**Figure 4. Provenance knowledge: An example with named graphs.**

```
CONSTRUCT (?Y recommendedFor ProvenanceWorkshopPC)
WITH PROVENANCE G3,G4
WHERE
{
  {
    { ?X collaboratesWith ?Y . } AND
    { ?X publishedOn ?Z .     }
  }
  FILTER ?Z = "ISWC"
}
FROM NAMED G1,G2
```

**Figure 5. Application scenario: Sample query.**

## Provenance challenges

The need for deeper understanding and use of provenance began to emerge with the growing popularity of distributed, large-scale collaborative environments and Semantic Web technologies. Today, this process is far from being complete. There are several promising research directions to leveraging provenance knowledge to its full potential.

### Interoperability

In many cases, data and knowledge are processed by different loosely coupled systems or workflows. The lack of consistent, coherent terminology for provenance-related concepts makes tracking, propagating, and querying provenance information difficult. A consistent terminology and representational standards would help outsiders easily grasp issues and compare systems. The interoperability of provenance components in collaborative environments is a key issue of the Provenance Challenges, which multiple research groups conduct worldwide.[17] The key idea of the Provenance Challenges is to better identify requirements and mechanisms

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns>
@baseURI http://www.problem.com/sampleQueryAnswer

:G11 {
        Bob recommendedFor ProvenanceWorkshopPC .
        Hugo recommendedFor ProvenanceWorkshopPC
      }

:G11Meta
      {
        :G11 source <http://example/report01.doc> ,
                    <http://example/image01.jpg> .
        :G11 agent Bob , Mary .
        :G11 extractor textAnalyzer ,
                       imageAnalyzer .
        :G11 certainty "0.71" .
        :G11 timestamp "6/6/2006" .
        :G11 how-provenance "(t1 ∧ t3) ∨ (t5 ∧ t4)"
      }
```

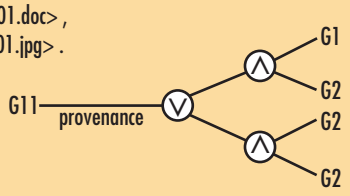


**Figure 6. Query results.**

for sharing provenance-related information across platforms, systems, and applications. The experimental framework typically describes sample data-processing workflows. It aims to systematically compare the following characteristics of different groups (with different research emphases, workflow execution environments, representation technologies, query languages, and so forth):

- how each system represents the workflow,
- how the given sample workflow represents provenance,
- how each system answers a set of predefined core provenance queries, and
- how each system represents the result of these provenance queries.

So far, the Provenance Challenges have resulted in early conceptual design recommendations for standardizing provenance knowledge for workflows.[17] However, this is only the first step toward a fully functional standard for provenance-aware workflow interoperability.

## Provenance beyond RDF

Understanding provenance mechanisms for relational databases[8] and Resource Description Framework repositories[10] can be a starting point for provenance-aware Semantic Web applications. Owing to the substantially higher complexity of inferencing and retrieval algorithms (such as reasoning in OWL-DL versus RDF querying with SPARQL) and the knowledge sources' distributed nature, the notion of why/where/how provenance will require further, nontrivial justification. Another interesting research issue is the support for *nested* provenance (that is, situations where provenance knowledge can have many different sources, time stamps, or degrees of reliability or belief).

## Time factor

If the data is fine grained and the provenance information is rich, provenance information can grow to be larger than the data it describes. For instance, the use of Named RDF Graphs for reification in an RDF repository could cause a large triple bloat: adding provenance knowledge could result in a tenfold (or more) blowup of the repository.[12] So, the manner in which the provenance metadata is stored and propagated between actors of the collaborative environment is crucial to its scalability.

Managing provenance incurs collection and storage costs. To reduce storage overhead, you can archive provenance information used infrequently or retain such provenance information in a demand-supply model based on usefulness. Alternatively, you can improve scalability by recording just the immediately preceding transformation step that creates the data and recursively inspecting the provenance information of those ancestors for the complete derivation history. However, in this case, there's a clear trade-off between storage costs and access time for recursively querying provenance knowledge through remote systems and workflows in the causal graph.

If provenance depends on users manually adding annotations instead of automatically collecting them, the burden on the user could prevent complete provenance from being recorded and available in a machine-accessible form that has semantic value.[18]

In this column, I raised the conceptual question of the Semantic Web layer cake's proof layer. I believe that provenance knowledge will be increasingly important at this abstraction level—and possibly the key to establishing trust mechanisms and advanced retrieval methods in decentralized applications. Work on Semantic Web provenance issues has started in parallel in different communities in the last decade. However, it's still far from being complete. Understanding provenance knowledge as a first-class citizen of distributed semantic workflows, knowledge bases, and inferencing and retrieval mechanisms will require further convergence of different viewpoints, standardization for seamless interoperability, and possibly new linking and querying methods for user-oriented navigation in the "Web of provenance." ■

## References


1. J.W. Murdock et al., "Explaining Conclusions from Diverse Knowledge Sources," *Int'l Semantic Web Conf.* (ISWC), LNCS 4273, Springer, 2006, pp. 861–872.
2. P. Pinheiro da Silva, D. McGuinness, and R. Fikes, "A Proof Markup Language for Semantic Web Services," *Information Systems*, vol. 31, no. 4, 2006, pp. 381–395.
3. D. McGuinness and P. Pinheiro da Silva, "Explaining Answers from the Semantic Web: The Inference Web Approach," *J. Web Semantics*, vol. 1, no. 4, 2004, pp. 397–413.
4. B. Aleman-Meza et al., "Semantic Analytics on Social Networks: Experiences in Addressing the Problem of Conflict of Interest Detection," *Proc. 15th Int'l Conf. World Wide Web* (WWW 06), ACM Press, 2006, pp. 407–416.
5. J.J. Carroll et al., "Named Graphs, Provenance and Trust," *Proc. 14th Int'l World Wide Web Conf.* (WWW 05), ACM Press, 2005, pp. 613–622.

6. L. Ding et al., "On Homeland Security and the Semantic Web: A Provenance and Trust Aware Inference Framework," *Proc. AAAI Spring Symp. AI Technologies for Homeland Security*, AAAI Press, 2005.

7. Y. Cui and J. Widom, "Practical Lineage Tracing in Data Warehouses," *Proc. 16th Int'l Conf. Data Eng.* (ICDE 00), IEEE Press, 2000, pp. 367–378.

8. P. Buneman, S. Khanna, and W.C. Tan, "Why and Where: A Characterization of Data Provenance," *Proc. 8th Int'l Conf. Database Theory* (ICDT 01), LNCS 1973, Springer, 2001, pp. 316–330.

9. N. Fuhr and T. Rölleke, "A Probabilistic Relational Algebra for the Integration of Information Retrieval and Database Systems," *ACM Trans. Information Systems*, vol. 15, no. 1, 1997, pp. 32–66.

10. S. Sizov, B. Schueler, and S. Staab, "Management of Meta Knowledge for RDF Repositories," to be published in *IEEE Int'l Conf. Semantic Computing* (ICSC 07), Sept. 2007.

11. M. Schraefel et al., "CS AKTive Space: Representing Computer Science in the Semantic Web," *Proc. 13th Int'l Conf. World Wide Web* (WWW 04), ACM Press, 2004, pp. 384–392.

12. J.J. Carroll and P. Stickler, "RDF Triples in XML," *Proc. Extreme Markup Languages Conf. 2004*; www.idealliance.org/papers/extreme/proceedings/xslfo-pdf/2004/Stickler01/EML2004Stickler01.pdf.

13. C. Bizer and J.J. Carroll, "Modelling Context Using Named Graphs," *W3C Semantic Web Interest Group Meeting*, Mar. 2004; http://lists.w3.org/Archives/Public/www-archive/2004Feb/att-0072/swig-bizer-carroll.pdf.

14. P. Groth, M. Luck, and L. Moreau, "A Protocol for Recording Provenance in Service-Oriented Grids," *Revised Selected Papers 8th Int'l Conf. Principles of Distributed Systems* (OPODIS), LNCS 3544, Springer, 2004, pp. 124–139.

15. P. Townend, P. Groth, and J. Xu, "A Provenance-Aware Weighted Fault Tolerance Scheme for Service-Based Applications," *Proc. 8th IEEE Int'l Symp. Object-Oriented Real-Time Distributed Computing* (ISORC 05), IEEE CS Press, 2005, pp. 258–266.

16. M. Szomszor and L. Moreau, "Recording and Reasoning over Data Provenance in Web and Grid Services," *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, LNCS 2888, Springer, 2003, pp. 603–620.

17. L. Moreau et al., "The First Provenance Challenge," *J. Concurrency and Computation: Practice and Experience*, 2007, pp. 1–7.

18. J. Zhao et al., "Semantically Linking and Browsing Provenance Logs for E-science," *Semantics of a Networked World*, LNCS 3226, Springer, 2004, pp. 158–176.

**Sergej Sizov** is a research assistant in the Computer Science Department at the University of Koblenz-Landau. Contact him at sizov@uni-koblenz.de.