

Support for Provenance in a Service-based Computing Grid

Shrija Rajbhandari and David W Walker
School of Computer Science, Cardiff University
PO Box 916, Cardiff CF24 3XF

Abstract

There is underlying need to support data provenance in a service-based computing environment such as the Grid where web services may be automatically discovered, composed, and then consumed using innovative workflow management systems. In many scientific scenarios a succession of data transformations occurs producing data of added scientific value. The provenance of such data needs to be recognized for other potential users to verify the data before using it in further studies. In this paper we present general requirements and implementation issues in provenance data collection, recording and reasoning, and discuss how these reflect on what aspects of information are essential for an effective and scalable system.

1 Introduction

In a large-scale, open and distributed environment web services may be automatically discovered, composed, and then consumed using innovative workflow management systems, thus allowing complex data processing of geographically distributed tasks. In such systems, large numbers of result data sets, in the form of data files and data parameters, are produced. As the number of such data sets increases, it may become difficult to identify and keep track of them. The problem of knowing precisely how a result data set was derived is of great importance in any large-scale scientific computation. Thus, it is necessary to capture the history of the derivation of result data sets and data products, which can then be used to provide an explanation of how such data products were produced. This historical and related metadata is usually referred to as ‘provenance’ which uniquely defines data and provides a traceable path to its origin. Provenance is particularly important when such data products are to be shared and/or re-used.

Electronically published scientific journal articles are the source of annotations or descriptions of the studies or experiments carried out. A scientific article is a part of the data provenance and moreover it also describes the output data’s provenance, for example the methods by which such data has been achieved. Provenance can be related in terms of input data and the resultant data associated with scientific processes. Typically, researchers maintain private records

of provenance in the file structure, databases or electronic notebooks which are in many cases sharable usually with other associated and authorized community members [1, 2, 3]. However, the scientific tools and programs used, and the processed and derived data sets, are normally not publicly available for use. Journal articles are the only public source for demonstrating studies revealing the origin of data and knowledge of how a particular result is obtained. We believe that such documents published online should be provenance-enabled, thereby allowing a reader of the document to access services to re-execute the work and view and evaluate the results produced on-the-fly without exposing the underlying knowledge of how they are re-created and where the data and programs are held. Thus, in this paper we describe a means of exploiting provenance data to enable the use of such an interactive document, which we have previously referred to as a “living document” [4]. We discuss the capture and recording of provenance information about scientific workflow processes, and tools to use this provenance data for re-creating and re-executing workflows. The nodes of these workflows are assumed to be services that transform their inputs into their outputs.

In a service-oriented approach to distributed computing application resources are regarded as services available on the network that, in collaboration, provide a comprehensive and flexible system solution [5]. Web service technology supports a Service-Oriented Architecture (SOA)

in which independent Web services can be discovered and dynamically bound across organisational domains. These individual web services can be composed together to form a composite service representing a workflow process which can then be enacted. Web service composition is an increasingly important theme in research into service-oriented architectures for Grid computing, as may be seen in projects such as GEODISE [6] and myGrid [7]. Web services are now becoming the basis for Grid computing, and middleware, standards, and tools for Web services are an active research area, with much effort being focused on the development of infrastructure for an adaptable and interoperable service-oriented framework for both scientific and business applications.

In a service-oriented Grid environment, the provenance of data consists of the entire processing history. This includes the identification of the origin of the raw data set, information on instruments that recorded the original data and parameters set, as well as all the processes that have been applied in the transformation of such data sets. Recording of such provenance information is evident in many applications and without provenance information it is impossible to trace and investigate any errors during data processing, or to assess the impact of faulty data sets and programs that has been used for data derivation.

As discussed in [8] there are two different granularities of provenance; course-grain and fine-grain provenance. Fine-grain provenance mainly refers to tracing the movement of data between databases [9] and to information on the ‘where’ and ‘why’ of a data item of interest that is the result of a database query [10]. Whereas, course-grain provenance refers to data sets generated through processing a workflow [11]. In this paper we are concerned with provenance in relation to workflow execution in a service-based infrastructure, i.e., course-grain provenance.

2 Related Work

The Chimera Virtual Data System is an important research product on data lineage [12]. Its architecture is based on the Chimera Virtual Data Schema where transformation elements describes programs, and the arguments describe data input/output. The main purpose of maintaining such a description in the Chimera virtual data catalog is for tracking how the data prod-

uct was created, what programs were used in transforming it, and to re-create the data product using such provenance information.

A primary work on provenance has also been performed in the myGrid project [13] where derivation provenance and annotations are recorded for *in silico* experiments performed using the myGrid laboratory workbench. Provenance is used to enable validation and re-creation of such experiments for other users as well as for the creator. In addition to this, the myGrid work has also pointed out that tracing the source of a derived object could verify the data quality and the trust issues in terms of the processing steps used [14]. The project purely focuses on the provenance in the area of bioinformatics specific to myGrid.

Another e-Science project called GEODISE [15] has developed a metadata manager which is used within a problem-solving environment (PSE) to store metadata about the data that the user has optimized or produced. It generates some standard metadata for the derived data files such as location, size and format and also allows the user to enter additional metadata, such as application-specific information [16]. The GEODISE model stores this metadata in the metadata database and the data files in the appropriate location, i.e., a file server. This allows the searching of data files without remembering the actual filenames and handles. In this scenario there is no indication of any automation in capturing information during the execution of the workflow processes, but manual annotation of complex data structures is supported which is then represented in a structured manner and stored in the metadata database.

The Earth System Workbench (ESSW) [3] project proposes a data management infrastructure which allows researchers to publish their large datasets from environmental models and global satellite imagery. In ESSW, a meta-model is created for tracking the documented experiments or lineage. The project has built a client-server based application called *Lab Notebook* where a scientist can define metadata templates in the form of XML DTDs to publish input data, derived data, programs, steps, etc. These are referred to as *science objects* and are specific to the ESSW project. The ESSW project will provide tools for finding, reusing and creating metadata.

A web-based tool called the *metadata manager* has been developed by the Council for the

Central Laboratory of the Research Councils (CCLRC) [17]. The tool allows authenticated users to enter metadata about a derived dataset such as location, purpose of study, etc. This tool is being used by some e-Science projects [18], such as *eminerals* [19], for recording metadata of derived data.

In [20] it is argued that the automation of data lineage collection and storage is essential, and need not affect current application practices. A basis for this argument is that manually entering information could lead to incomplete documentation of provenance.

The proposed Scientific Annotation Middleware (SAM) project [21] is primarily a middleware infrastructure with a set of layers of services and components to support the generation, storage, searching, translation, and navigation of scientific metadata, semantic relationships of data lineage, notebook-specific metadata, and relationships. The aim of this project is to enable the sharing of scientific records among portals and problem-solving environments, software agents, scientific applications, and electronic notebooks. Its components are being developed in close corporation with domain-oriented SciDAC¹ application projects (e.g., climate modelling, computational chemistry, etc) and relevant technologies from the SciDAC Grid. CMCS (the Collaboratory for the Multi-scale Chemical Sciences) research project is one of the projects using SAM for their pedigree implementation [22]. The CMCS project has developed tools for viewing provenance relationships and for using them to do scope notifications and searches. It provides flexible methods to track data provenance across many disciplines and also supports multiple uses of provenance [23].

In [24] a client-server based infrastructure for supporting provenance recording and querying is proposed. This is primarily based on creating a structured database system, with a set of methods for recording and querying provenance data in the database. In addition, results can be recreated based on the queried provenance. The paper also provides an overview of data provenance issues, such as trust.

3 Provenance Model

The incorporation of provenance capabilities into a service-based Grid environment requires two main functionalities:

1. The ability to collect and archive the provenance about the transformation of data occurring during invocation of web services, for example, a composite service executed via a workflow engine.
2. Allowing such recorded provenance to be accessible and viewable via generic browsers and query interfaces.

In this paper a provenance model for Grid and Web service architectures is presented which focuses on the requirements of the provenance data. For example, how is the provenance data collected, represented and recorded, and how is the provenance data queried and reasoned with.

In this paper, we focus only on the provenance of a derived data set that is a result of a composite service executed via a workflow engine. The collected and recorded provenance can be represented either as a composite service itself that can be exposed as a Grid or Web service, or just as metadata that is associated with a particular derived dataset for tracing its origin. Provenance is recorded in a repository that can be accessed via tools for utilizing and manipulating provenance.

Figure 1 illustrates our provenance-enabled service oriented architecture. The architecture shows a provenance server that provides two services, the Provenance Collection Service (PCS) and a Provenance Query Service (PQS). The PCS is responsible for collecting all the details of the transformation of data sets occurred during the workflow execution and recording them in an appropriate provenance database. Provenance is recorded so that a workflow could be re-executed based on its given provenance. We assume that the workflow engine considers the PCS as a trustable resource for provenance recording.

A number of software systems are available to facilitate the manual composition of web services and more importantly the execution of such composed web services. Also many workflow description languages have been proposed [25, 26]. Business Process Execution Language (BPEL4WS) is one of the most popular workflow description languages and provides a mechanism to specify the structure of a workflow. BPEL compositions rely heavily on WSDL descriptions of the invoked services in order to refer to the messages being exchanged, the operations being invoked, and the port types these operations belongs to. Thus, BPEL is build on top of

¹SciDAC = Scientific Discovery through Advanced Computing (<http://www.osti.gov/scidac/>)

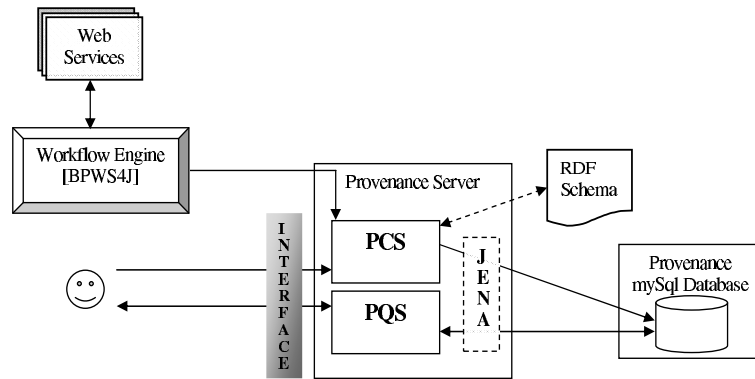


Figure 1: Provenance Model

WSDL. To our knowledge the two workflow execution engines available which provides tools for executing BPEL workflow description are; IBMs Business Process Execution Language for Web Services Java Run Time (BPWS4J) [27] and BPEL Orchestration Server - Collaxa [28]. The former is a closed-source distribution but very popular and widely used, whereas the latter is a commercial product. We use the BPEL workflow description language to compose and describe workflows, and BPWS4J for an enactment engine. The BPWS4J engine does the necessary invocation of all the related web services by executing the BPEL document that describes the execution order of the services, and the related data flow and control flow. We record provenance data of the composite workflow itself, including the workflow identifier and detailed description. In addition, for each job executed we record details of the input and output datasets in the provenance repository. Provenance is represented by means of Resource Description Framework (RDF) [29] with a pre-defined Provenance RDF Schema and recorded in a mySql database.

In order to retrieve and utilize the provenance data, we provide tools for searching, visualizing and verifying provenance in the database. A set of web pages are provided to allow a workflow to be re-run based on the queried provenance data. This helps to update the provenance database. For example, if repairs or revisions are made to any of the services and/or input data that were used originally; this will make the workflow's provenance record worthless. Thus, in such a scenario re-execution updates the provenance database with correct provenance data. Also, re-execution with a

given provenance helps to discover if any services have evolved or are unavailable based on the given input and output data set. Thus, an unchanged output data set with the same initial input data set and the same service indicates a service that has been unchanged.

3.1 Provenance Ontology, Collection and Recording

The BPWS4J enactment engine exposes a complex service described in the BPEL language as a single web service. The location of this service description, the input parameters and their descriptions are feed in via our interface to the Provenance Collection Service (PCS). The PCS acts as a client for initiating the invocation of the web service and sends the input parameters to the BPWS4J engine for execution. The PCS recording starts when it receives the necessary arguments, such as the location, exposed via BPWS4J, of the composed service description, and the input parameters. It creates a new workflow identifier for each workflow in the database and records:

1. the syntactic metadata of the services from the service description provided,
2. the input parameters passed as arguments,
3. other metadata such as timestamps.

Then, the service exposed by the BPWS4J engine is initiated by PCS, which invokes all its constituent services according to the data and control flow in its BPEL document, and then returns the result data to the PCS. Finally, the PCS records the output data in the provenance database.

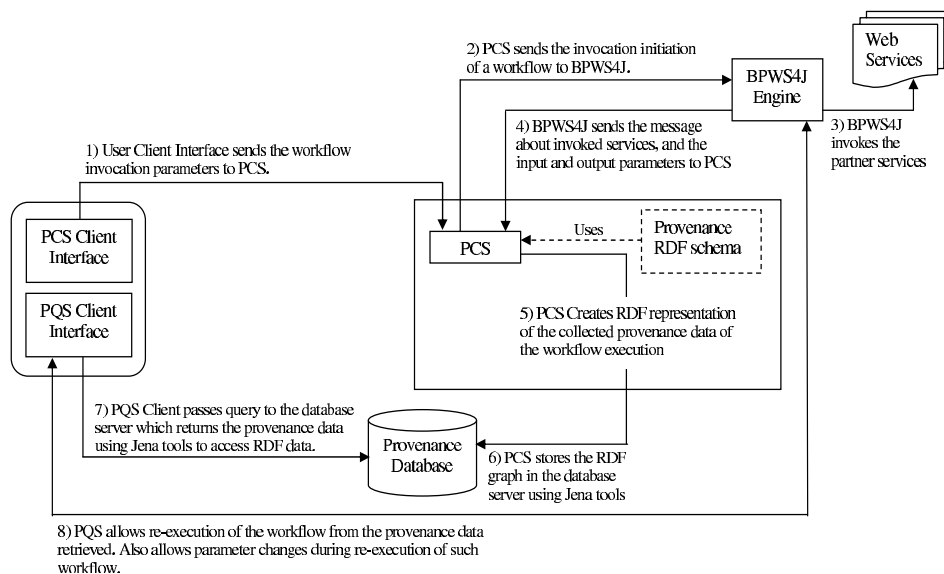


Figure 2: Interaction between the PCS, PQS Client and BPWS4J

The syntactic metadata, captured by the PCS, of the derivation history of the workflow execution are output as an RDF statement graph in the form of *subject-predicate-object* tuples. RDF graphs are then stored in a MySQL database using HP's Jena toolkit [30], which enables the creation of RDF models and statements and also structures the database. Pre-defined provenance RDF schemas, *WorkflowProvenance.rdfs* and *Profile.rdfs*, are used by the PCS to structure the provenance of the workflow trace in RDF format. *WorkflowProvenance.rdfs* is used to structure the constituent services of which the workflow is composed, and also properties of the workflow, such as its description, start-time and end-time, identifier, and user ID. *Profile.rdfs* is used to structure details of the constituent services, such as the input and output datasets, operations, and the WSDL location of services. An RDF schema provides semantical information about the interpretation of the statements given in an RDF data model.

RDF and RDFS provide an elementary ontology language that not only provides a data model in XML but also enables the representation of classes, properties, domain and range and sub-class and super-class hierarchies. The provenance of a workflow represented by RDF can be viewed and exposed as a Grid or Web service. It is an advantage when the provenance of every Web service that was invoked is represented in RDF when this is coupled to a higher

level service ontology. Using an ontology provides a shared conceptualization based on semantic proximity of terms in a specific domain of interest that facilitates searching and discovery of Web service provenance [31]. The emerging Semantic Web makes it possible to specify semantics of a domain, such as the terms and concepts of interest, their meanings, relationships between them and the characteristics of the domain through an ontology [31].

The PCS enables workflow provenance to be recorded using the Web Service Invocation Framework (WSIF) [32] for invocation of the workflow Web service. The input and output data types describing the service can be either simple types, like integer and string, or a complex structure of such simple types. The challenge was to be able to handle the complex types returned from the invoked service. Adopting the Java Record Object Model (JROM) [33] and representing the input and output message as JROM values can effectively handle the complex types of the returned output. JROM also enables the conversion of JROM representation to XML, and visa versa. This enables to structure the complex type accurately while representing it in RDF statements. JROM can be used to handle simple and complex schema definitions, but currently arrays of complex types are not supported.

3.2 Reasoning and Exploiting Provenance Information

A form-based provenance query service (PQS) is provided that allows a user to query the database and browse the RDF statements of provenance information for any workflow. The recorded provenance in RDF is queried using the Jena tools through the PQS interface. Here we are not concerned about the structural outline of the provenance records in the database as long as they are retrieved effectively. The PQS interface allows viewing of provenance data either as an XML-RDF document or in RDF statement format. The provenance retrieved by PQS includes location of all the invoked constituent services, their input and the output data details, timestamp, and so forth. A set of web pages with a *reinvoke()* method in PQS enables re-execution of previously run workflows using the retrieved provenance data from the database. Re-execution of workflow here assists in verification of the queried provenance, which is done simply by matching the input and output data of the re-executed result with the original provenance input and output data. Based on this simple checking method it could be said that the provenance data can be kept up to date. The tools also allow the user to choose to change the input parameters of a re-run workflow to enable a ‘what-if’ style of investigation.

4 Conclusions

Starting with the initial motivation of the “living document” concept for interactive online publication, in this paper we have presented a simple framework for exploiting provenance capabilities in a service-oriented Grid, which mainly focuses on the execution of composite Web services through a workflow engine. As mentioned earlier, the two main functionalities in the provenance model are recording and reasoning about workflow provenance. The recording of workflow trace provenance is possible in two ways: synchronous and asynchronous collection. The synchronous way that we have adopted in our current prototype system for extracting provenance from the workflow engine means that once the workflow execution starts, the recording program is prevented from doing any processing until the current execution completes. This approach is useful for simple workflows that execute in a short time and provides a complete

workflow trace log file after completion of the execution.

On the other hand, in the asynchronous approach, both the workflow engine and the recording programs run in parallel. This allows the frequent submission and recording of details of each service as it is invoked. This is desirable particularly for larger workflows handling large datasets in a scientific computation. As such scientific processes are often extremely complex and time consuming, any intermediate loss of data or anomalies could be costly, so that synchronous provenance recording could be inefficient. With the use of asynchronous recording, in case of any failure or error, provenance can be used to trace back and investigate the error points and continue the remaining process execution.

Standardization of complex workflow trace metadata is an issue as there is no agreed standard way of representing such provenance data in the current infrastructure, such as the Dublin core metadata for the digital library community [34]. It is understandable that the requirements of process provenance vary across domains but the Dublin core standard could be used to describe workflow provenance data. Other issues, such as trust in relation to the source of the provenance data describing the original dataset, as well as the Web service or the programs used, is very open and widely recognised [24]. Trust issues could be minimal in cases where provenance management is domain specific and involves only authenticated and authorised users.

Recorded provenance is of great importance in scientific computation in which large amounts of data are processed for days or months. In this cases, the recorded provenance allows the user to reuse or analyze the resultant dataset without having to run the entire workflow process again. Reasoning about provenance data as offered by our existing prototype system should not be limited to just simple browsing, viewing and re-running the transformations, with identical or altered input parameters and data sets – it should be applied in other scenarios. For example, it is possible that some Web services could be unavailable, altered, or substituted. Till now the description of a Web service has been limited to the input and output messages, or also it could be semantically defined. But information such as ‘how long the service will be available for use or valid for’, i.e., “valid until” date, is not mentioned in service descriptions like WSDL. If

such information exists it can provide valuable information to update the workflow provenance data. Also, unavailability of any service could lead to failure in re-execution of the workflow. Thus, while wanting to re-run the same workflow from the provided provenance with altered or new datasets, it is desirable to have tools in the provenance system that can search and match semantically equivalent Web services, and then replace the unavailable service with the semantically identical one. This semantic matching capability would also be advantageous for checking the validity of provenance data during re-execution of the workflow of the given provenance. Applying the simple input output similarity matching that we adopt currently could be misleading as many scientific services produce random results every time when re-running with the same input datasets. Thus, syntactically non-identical results could be semantically identical.

References

- [1] Chin, G., Leung, R., Schuchardt, K., Gracio, D.: New paradigms in problem solving environments for scientific computing. In: 7th International Conference on Intelligent User Interfaces. (2002)
- [2] Myers, J.: Collaborative electronic notebooks as electronic records: Design issues for the secure electronic laboratory notebook (eln). In: International Symposium on Collaborative Technologies and Systems (CTS'03). (2003) 19–24
- [3] Frew, J., Bose, R.: Earth science workbench: A data management infrastructure for earth science products. In: 13th International Conference on Scientific and Statistical Database Management. (2001)
- [4] Walker, D., Rana, O.F., Li, M., Shields, M.S., , Huang, Y.: The software architecture of a distributed problem-solving environment. *Concurrency: Practice and Experience* **12** (2000) 1455–1480
- [5] Layman, F.: Web services: Distributed applications without limits- an outline-. In: The First International Conference on Service Oriented Computing. (2003)
- [6] Geodise: Geodise. <http://www.geodise.org/> (2004)
- [7] myGrid: mygrid. <http://www.mygrid.org.uk/> (2004)
- [8] Workshop, P.: Data provenance and annotation workshop. <http://www.nesc.ac.uk/esi/events/304/> (2003)
- [9] Buneman, P., Khanna, S., Tan, W.: Data provenance: Some basic issues. In: Foundations of Software Technology and Theoretical Computer Science (FSTTCS). (2000)
- [10] Buneman, P., Khanna, S., Tan, W.: A characterization of data provenance. In: International Conference on Database Theory (ICDT). (2001)
- [11] Wilde, M.: Survey session: Course grained provenance. In: presented at Data Provenance and Annotation Workshop. (2003)
- [12] Foster, I., Vockler, J., Wilde, M., Zhao, Y.: Chimera: A virtual data system for representing, querying, and automating data derivation. In: 14th International Conference on Scientific and Statistical Database Management. (2002) 37–46
- [13] Addis, M., Ferris, J., Greenwood, M., Marvin, D., Li, P., Oinn, T., Wipat, A.: Experiments with e-science workflow specification and enactment in bioinformatics. In: UK e-science All Hands Meeting. (2003) 459–467
- [14] Greenwood, M., Goble, C., Stevens, R., Zhao, J., Addis, M., Marvin, D., Moreau, L., Oinn, T.: Provenance of e-science experiments - experience from bioinformatics. In: UK e-science All Hands Meeting. (2003) ISBN 1–904425–11–9
- [15] Cox, S., Jiao, Z., Wason, J.: Data management services for engineering. In: UK e-Science All Hands Meeting. (2002)
- [16] Wason, J., Molinari, M., Jiao, Z., Cox, S.: Delivering data management for engineers on the grid. In: Euro-Par 2003 Parallel Processing, Lecture Notes in Computer Science. (2003) 413–416
- [17] CCLRC: Council for the central laboratory of the research councils. <http://www.cclrc.ac.uk/Home/> (2004)
- [18] e Science centre, C.: Cclrc. <http://www.e-science.clrc.ac.uk/web/> (2004)

- [19] e minerals: e-minerals project site. <http://eminerals.esc.cam.ac.uk/eminerals/> (2004)
- [20] Boss, R.: A conceptual framework for composing and managing scientific data lineage. In: 14th International Conference on Scientific and Statistical Database Management. (2002) 47–55
- [21] SAM: Scientific annotation middleware. <http://collaboratory.emsl.pnl.gov/docs/collab/sam/> (2004)
- [22] csm: Computational science and mathematics projects site. <http://csm.pnl.gov/sce/projects.stm> (2004)
- [23] Myers, J., Pancerella, C., Lansing, C., Schuchardt, K., Didier, B.: Multi-scale science: Supporting emerging practice with semantically derived provenance. In: Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data. (2003)
- [24] Szomszor, M., Moreau, L.: Recording and reasoning over data provenance in web and grid services. In: International Conference on Ontologies, Databases and Applications of SEMantics (ODBASE 03). (2003)
- [25] WSFL: Web services flow language. <http://www-306.ibm.com/software/solutions/webservices/pdf/WSFL.pdf> (2004)
- [26] BPEL4WS: The ibm business process execution language for web services. <http://www-106.ibm.com/developerworks/library/ws-bpel/> (2004)
- [27] bpws4j: The ibm business process execution language for web services java. <http://www.alphaworks.ibm.com/tech/bpws4j> (2004)
- [28] Collaxa: Collaxa bpel server. <http://www.collaxa.com/product.welcome.html> (2004)
- [29] RDF: Resource discription framework. <http://www.w3.org/RDF/> (2004)
- [30] Jena: Jena toolkit. <http://www.hpl.hp.com/semweb/tools.htm> (2004)
- [31] Sheth, A., Bertram, C., Avant, D., Hammond, B., Kochut, K., Warke, Y.: Semantic content management for enterprises and the web. In: IEEE Internet Computing. (2002) 80–87
- [32] WSIF: Web services invocation framework. <http://ws.apache.org/wsif/> (2004)
- [33] JROM: Java record object model. <http://www.alphaworks.ibm.com/tech/jrom> (2004)
- [34] dublinCore: Dublin core metadata standard. <http://dublincore.org/> (2004)