Future Generation Computer Systems 26 (2010) 324–336

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

An ActOn-based semantic information service for Grids

Wei Xing [a,*], Oscar Corcho [b], Carole Goble [a], Marios D. Dikaiakos [c]

[a] Information Management Group, School of Computer Science, University of Manchester, UK
[b] Ontology Engineering Group, Departamento de Inteligencia Artificial, Facultad de Informática, Universidad Politécnica de Madrid, Spain
[c] Department of Computer Science, University of Cyprus, Cyprus

ARTICLE INFO

Article history:
Received 20 September 2008
Received in revised form
13 September 2009
Accepted 8 October 2009
Available online 15 October 2009

Keywords:
Information services
Ontology
Semantic Grid
Grids

ABSTRACT

We describe a semantic information service that aggregates metadata from a large number of information sources of a large-scale Grid infrastructure. It uses an ontology-based information integration architecture (ActOn) suitable for the highly dynamic distributed information sources available in Grid systems, where information changes frequently and where the information of distributed sources has to be aggregated in order to solve complex queries. These two challenges are addressed by a Metadata Cache that works with an update-on-demand policy and by an information source selection module that selects the most suitable source at a given point in time. We have evaluated the quality of this information service, and compared it with other similar services from the EGEE production testbed, with promising results.

1. Introduction and motivation

Grid infrastructures integrate large computational and storage resources, data, services and applications from different disciplines [1,2]. For example, EGEE [3] provides a production quality grid infrastructure spanning more than 30 countries with over 200 sites to a myriad of applications from various scientific domains, including Earth Sciences, High Energy Physics, Bioinformatics and Astrophysics. It currently includes over 35 000 CPUs, 13 Petabytes of storage space in hundreds of storage elements, and an average of 40 000 concurrent jobs per day on behalf of 100 Virtual Organisations (VOs). Another example is the Open Science Grid (OSG) in USA, which includes over 15 000 CPUs, 3 Petabytes of storage space, and an average of 16 000 jobs per day on behalf of 48 VOs.

The ability to manage and operate such large-scale Grid systems depends on the availability and accuracy of information about individual domains, about capabilities of resources, such as computational power, storage, networking, and sensing, and about complex interconnected behaviours between systems. Such information is normally provided by information services available in these infrastructures. For example, in an infrastructure like EGEE, examples of these deployed information services are BDII [4] and MDS2 [5], which capture information about hardware and software resources, and RGMA [6], which is focused on jobs, services and running environments. And in Open Science Grid, available information service are GIP (Grid Information Provider) and CEMon. GIP provides information about a site on the OSG Grid. The CEMon services publish information about computing elements the OSG Grid.

The main limitations of existing information services are that they do not provide enough information about large-scale distributed systems, since they only focus on a few specific aspects of such systems, and that they do not always provide accurate information about the actual status of the Grid resources that they refer to. As aforementioned, BDII [4] and MDS2 [5] capture information about hardware and software resources, but do not provide information about data sources, networking connections, services and running environments. Furthermore, in some cases the information models used by existing information services are ill-defined or cannot be handled easily to solve general-purpose queries. For example, in MDS4 [7] the keyword "MPI" is used to describe that a site is "MPI-enabled", but this does not necessarily mean that the MPI configuration is ok in that site, what is missing from that information model. Our experience in Crossgrid, LCG Grid, Open Science Grid, and EGEE Grid shows that this can lead to failures or inadequate behaviours in other middleware services that heavily depend on information services, like resource brokers, job schedulers, etc.

To overcome these limitations, we propose the creation of a semantic information service that integrates information from different information sources according to a rich ontology-based information model. The integration of distributed information poses the following *challenges*, due to the dynamic and heterogeneous nature of Grids:

* Corresponding author.
E-mail addresses: wei.xing@manchester.ac.uk, xingwei2006@gmail.com (W. Xing), ocorcho@fi.upm.es (O. Corcho), carole.goble@manchester.ac.uk (C. Goble), mdd@cs.ucy.ac.cy (M.D. Dikaiakos).

0167-739X/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2009.10.003

- Metadata of a Grid entity consists of multiple attributes, whose values can be normally obtained from heterogeneous and geographically-distributed information sources. In a large-scale Grid system, several information sources can provide the same piece of information about a resource. And it may be difficult to identify and locate the most suitable (and available) information source for a specific information need.
- Metadata about most Grid entities may be updated frequently, so as to reflect the current status (capability and availability) of the services and resources that it refers to. This makes it hard to create and maintain up-to-date metadata about all the resources available in a Grid. For instance, the usage level of a CPU, storage space, and network connection may change every few minutes.
- Different information sources or services may provide overlapping views of the Grid state, in different schemas and formats, and with different characteristics of their information provenance (update frequency, quality-related).

Ontology-based information integration [8] is one of the approaches that has been traditionally used to address these challenges, creating information services that can be used to query multiple information sources transparently. Among the existing ontology-based information integration approaches we have those that access information sources and transform information into a common format on demand (that is, when information requests are sent to the system), and those that retrieve and consolidate information using batch processes that are executed at regular time intervals (normally due to the fact that information extraction or aggregation is time-consuming because of its complexity or because additional curation steps are needed). These are normally known as virtualised and materialised views respectively. However, none of the approaches that we have analysed is adequate in the dynamic, large-scale distributed setting described above, due to the following *limitations*, which will be explained in detail in Section 2.

- *Ontology-based information integration systems are not prepared for highly dynamic information sources*. These systems assume that the data stored in the information sources does not change as frequently as is the case in Grid systems. Namely, the information is assumed to be *valid* for a long time, more than what is needed to execute the query and aggregate the information. This assumption cannot be taken for granted in a Grid system. In Grids, there are many time-sensitive resources and services, which change very frequently and with different timescales. For example, the usage of CPU resources, the status of job queues and network connections, and the storage space may change in minutes; the stability of services may change in hours; and the information about membership to a virtual organisation may change in days.
- *Ontology-based information integration systems are not fault-tolerant and robust to changes in the information source availability*. Most of these systems assume that there is only one information source available for each piece of information required, and that this information source is always available. In other words, most of these systems are configured at the design time so as to fetch information from a specific set of information sources, and in the case that one of the information sources is unavailable, they normally get stalled in their retrieval process or give back incorrect or incomplete information to their requestors. As mentioned earlier, in a large-scale distributed system duplication of information is common, hence there may be many geographically-distributed information services available for the same piece of information, with different service quality and cost. Hence, robust fault-tolerant integration systems should be able to select the most suitable information source, according to their preferences and to the information source status. Besides, traditional systems cannot easily adopt a new information source at run-time.

Our proposal to address all these challenges is to use ActOn (Active Ontology) [9], an ontology-based information integration system that especially focuses on the generation and maintenance of up-to-date metadata for dynamic, large-scale distributed systems.

As in other ontology-based information integration systems, ActOn uses ontologies to describe the domain for which information will be aggregated. This provides an expressive model to describe that information, which can be exploited with query languages and used for validation purposes (e.g., to detect inconsistencies in the aggregated information) and for deriving new information. It also provides an extensible data model where changes in the descriptions of resources and services, or in the information sources (update frequency, information quality, etc.) are automatically reflected in the behaviour of the system.

The main feature of ActOn is that it incorporates the following two modules to deal with the dynamicity of information and with changing information sources: a cache, which provides fast access to information that has been already integrated and materialised and which is still valid, and an Information Source Selector, which is used during the generation of the execution plan for retrieving information from the information sources and allows the system to adapt to changing conditions of the infrastructure and to add new information services easily. These modules are not always considered in existing information integration approaches.

The remaining of this paper is organised as follows. Section 2 discusses related work, focusing on the main similarities and differences between ActOn and other ontology-based information integration approaches described in the literature, and describing other Grid information services, used in the evaluation and comparison of our implementation. Section 3 presents the architecture of ActOn, focusing on its different software and knowledge components, and on the main interactions between them. We describe the prototype implementation for the EGEE Grid, which instantiates this architecture. Section 4 describes our evaluation, which covers aspects related both to quality and performance of our approach, and the obtained results, and compares them to those of similar Grid information services. Finally, Section 5 provides conclusions, and describes open issues and our planned future work.

2. Related work

In this section we will review related work in ontology-based information integration, in the development of Grid information services, and in the Semantic Grid. We will start with the former.

2.1. Ontology-based information integration systems

Many sets of criteria have been used for the classification of existing ontology-based information integration systems [8,10]. One sample criterion is the place where information resides, which allows us to distinguish between mediator and warehouse approaches [11], also known as virtual/on-demand and materialised/cache approaches. Another example is the distinction between systems using a single ontology, multiple ontologies, and hybrid approaches with shared and non-shared ontologies. Other works distinguish between the Local as View (LaV) [12–14] and the Global as View (GaV) [15] approaches. Others focus on the degree of automation of mappings between sources and ontologies [10].

These studies concentrate mainly on the technical aspects of each approach. However, we can also consider other important challenges that appear in information integration, some of which are described in [16]:

- Identity reconciliation. Recognising when different objects at different information sources denote the same entity.
- Efficient querying over the distributed information, which usually involves:

[*] Effective query reformulation & query planning.
[*] Query accounting, which considers the cost of querying an information source and avoids querying multiple times about the same piece of information.

- Information source selection.
- Legacy data transformation into semantic representations (that is, wrapper generation).

Fig. 1 shows how some of the most relevant ontology-based integration approaches take into account all of these features. We have selected the following approaches:

- SIMS [17], and its successor Prometheus [18], are on-demand approaches focused on integrating data from many different types of information source, including HTML pages, images, databases, etc. These approaches are strong in the query reformulation and planning techniques that they use for their mediation tasks (the planning is done by Theseus [19]). Prometheus also addresses identity reconciliation.
- Carnot [20], and its successor InfoSleuth [21], are on-demand approaches focused on integrating data from databases, although they could be easily extended to other types of information sources. The latter uses an agent-based paradigm to distribute the processing of queries among resource agents, which have previously advertised their capabilities in order to allow for a dynamic source selection. Both approaches propose techniques for query planning and identity reconciliation (data quality).
- TSIMMIS [15] and Information Manifold [14] are some of the early approaches to ontology-based information integration, addressed mainly to structured information sources such as databases. They are both on-demand approaches, with some form of query planning techniques. The first one is specially focused on the automatic generation of wrappers.
- OBSERVER [22], PICSEL [12] and TAMBIS [23] are similar on-demand access approaches that transform queries expressed in different description logic languages, with different expressiveness, into distributed queries over a set of information sources, which range from databases to semi-structured files in different formats (HTML, XML, etc.), and even services in the case of the latter. PICSEL (in its third version) includes a data warehouse for information that does not change.
- DWQ [24] is one of the few approaches focused on data warehousing. An important part of this approach is ensuring the quality of data in the data warehouse, hence different types of data quality techniques are applied. Here also the aspects related to the cost of accessing information sources are considered.
- KnowledgeParser [25] is also aimed at generating a knowledge base from the information available in different sources. Since it is mainly focused on unstructured and semi-structured sources, many hypotheses have to be taken into account in order to generate the knowledge bases, and the process is slow, not being suitable for cases where the information sources change frequently and where an on-demand access is needed.

The results shown in Fig. 1 allow reasserting our initial assumption about the fact that *none of the existing approaches is prepared for working on highly dynamic environments* (the pure on-demand approaches are too slow for providing results that take into account the frequency of changes in information sources, and the data warehouse approaches do not refresh their materialised information fast enough).

Besides, *only a few approaches are able to select dynamically from a set of overlapping information sources*, and in these cases the selection is never based on non-functional requirements such as the ones that we take into account, but only on logical conditions based on the information that they contain. Furthermore, the cost of sending the same queries frequently to the same information sources is not considered by most of the approaches.

At the same time, the information provided in the table shows that in our future work we can benefit from the large amount of work devoted to query reformulation and planning, and identity reconciliation, which could be useful when applying our approach to other scenarios.

2.2. Grid information services

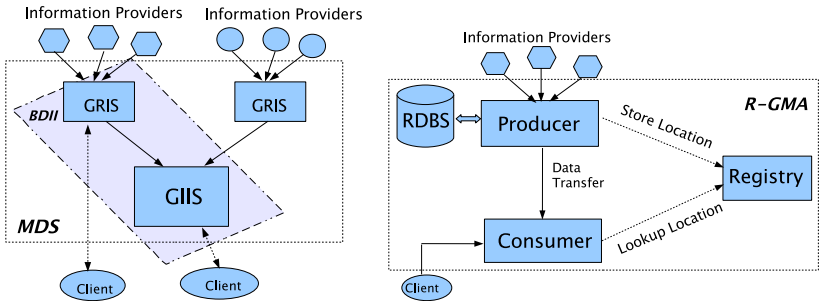
Now we move into Grid information services. Currently, there are three well-known Grid information services: Monitoring and Discovery System (MDS), Berkeley DB Information Index (BDII), and RGMA. These services are deployed in most Grid systems, such as Europe Data Grid, Crossgrid, NASA Grid, and Open Science Grid, and widely used by middleware and applications running on them. In the context of the EGEE Grid, where our information service is deployed, BDII and RGMA are adopted as the default information services.

Let us now describe these information services in detail.

- The Monitoring and Discovery System (MDS) is the information service component of the Globus platform. There are several versions of this system available and deployed in different infrastructures, being MDS-2 [5] and MDS4 [26] the most relevant ones. In MDS-2, information about Grid resources is extracted by "information providers", which are software programs that collect and organise information from individual Grid entities, either by executing local operations or by contacting third-party information sources (e.g., the Network Weather Service, SNMP, etc.). Extracted information is organised according to the LDAP (Lightweight Directory Access Protocol) data model in LDIF format and uploaded into LDAP-based servers of the Grid Resource Information Service (GRIS). GRIS servers can register themselves in the Grid Index Information Services (GIIS) in order to aggregate directories, using a soft-state registration protocol called Grid Registration Protocol (GRRP), as shown in Fig. 2(a). Moreover, a GIIS can register with other GIIS's, thus creating a hierarchy of aggregate directory servers. End-users can address queries to GIIS's using the GRIP protocol.
- MDS-4 provides standard interface to different information sources, translating their diverse schemas into appropriate XML schema. It is implemented as two WSRF-based services, the Index service and the Archive service. Compared to MDS2, MDS4 adopts a web service interface for information access, and XML as its data model for information representation. The query language for MDS-4 is Xpath, which is executed against the Resource Property Set of Grid resources.
- The BDII (Berkeley DB Information Index) [4] is an improvement of MDS that caches information using the Berkeley DB, using the same information model and access API. Like MDS, BDII is based on LDAP servers. It consists of two or more standard LDAP databases that are populated by an update process. The update process obtains LDIF either by doing an ldapsearch on LDAP URLs or by running a local script that generates LDIF. The LDIF is then inserted into the LDAP database.
- RGMA [6] is a framework that combines monitoring and information services based on a relational model that is implemented with XML. It has been built in the context of the EU DataGrid project and implements the Grid Monitoring Architecture (GMA) proposed by the Open Grid Forum. As shown in Fig. 2(b), GMA models the information infrastructure of the Grid using three core types of component: (i) producers, which provide information; (ii) consumers, which request information; and (iii) a single registry, which mediates the communication between producers and consumers. R-GMA implements

	General Approach		Information Source Wrapping			Mediation			
	Information Access Approach	Mapping Approach	Information Source Selection	Type of Legacy Data	Wrapper Generation	Query language & Query reformulation	Query Planning	Query Accounting	Identity reconciliation
TSIMMIS	On-demand Access	Local as view	Pre-defined	Structured data DBs	Automatic	SQL-type: LOREL	Yes	No	Yes
Information Manifold	On-demand Access	Global as view	Pre-defined	Structured data DBs	Manual	Predicates	Yes	No	No
Carnot	On-demand Access	Global as view	Pre-defined	DBs	Manual	SQL	Yes	No	Yes
InfoSleuth	On-demand Access	Local as view	Dynamic	DBs	Manual	OKBC-based	Yes	No	Yes
SIMS	On-demand Access	Global as view	Dynamic	Structured data DBs	Manual	DL: LOOM	Yes	No	No
Prometheus +Theseus	On-demand Access	Global as view	Pre-defined	HTML, Images DBs	Semi-automatic	Predicates	Yes	No	Yes
OBSERVER	On-demand Access	Local as view	Pre-defined	DBs, XML bib, HTML	Manual	DL: FLON	No	No	No
TAMBIS	On-demand Access	Global as view	Pre-defined	DBs, XML, HTML	Manual	DL: GRAIL	Yes	No	No
PICSEL	On-demand Access + Data Warehouse	Local as view	Pre-defined	Dbs, XML, HTML	Semi-automatic	DL: CARIN	Yes	No	Yes
DWQ	Data Warehouse	Local as view	Pre-defined	DBs	Manual	Datalog	No	Yes	Yes
Knowledge Parser	Data Warehouse	Local as view	Pre-defined	HTML, pdf, Word, DB	Semi-automatic	--	No	No	Yes
Active Ontology	On-demand Access + Data Warehouse	Global as view	Dynamic	Structured data DBs	Manual	SPARQL	No	Yes	No

Fig. 1. Features of the most common ontology-based information integration approaches.



(a) Overview of MDS and BDII. (b) The RGMA architecture.

Fig. 2. MDS, BDII and RGMA.

two additional properties over GMA. First, consumers and producers handle the registry in a transparent way; thus, anyone using R-GMA to supply or receive information does not need to know about the registry. And second, all the information appears as one large relational database and can be queried as such (anyway, in the current implementation, the database is centralised). R-GMA can be accessed using the R-GMA APIs.

Table 1 shows the difference among these Grid information services, including our prototype implementation. We compared the four wide-deployed Grid information services with the information model, information access protocol, and architecture aspects.

2.3. Semantic Grid and Knowledge Grid

The Semantic Grid is an initiative whose main foundation is the exposure of semantically rich information (metadata) associated with Grid resources, with the aim of exploiting it to build added-value Grid services. Metadata is exposed as a first class object in a machine processable form. One of the approaches to deal with metadata has been proposed by the Semantic Open Grid Service Architecture (S-OGSA), which extends OGSA by defining a lightweight mechanism that allows for the explicit use of semantics along the associated knowledge services to support a spectrum of service capabilities [27,28]. Another approach is that of the Knowledge Grid [29–31], which also acknowledges the need for explicit management of knowledge resources and metadata in Grids.

In the context of Semantic Grid research, several efforts have focused on the integration of metadata from different Grid resources or in the development of better information services than those currently available in Grid middleware and deployments [32]. Works like the one described in [33] are focused on how to use RDF in different application domains, by performing the integration of various information sources. In [34], the focus in on how to

Table 1
Features of the most common Grid information services.

Name	Data model	Query language	Architecture	Semantic	Metadata cache	Information access
MDS-2	LDAP Tree	LDAP	Distributed	No	No	On-demand
MDS-4	XML	XPath	Distributed	No	No	On-demand
BDII	LDAP Tree	LDAP	Centralized	No	Yes	Data Warehouse
RGMA	Relational	SQL	Distributed	No	No	On-demand
ActOn	RDF Graph	SPARQL	Centralized	Yes	Yes	On-demand + Data Warehouse

provide semantic descriptions of services. In both cases, the main assumption is that information sources do not change frequently. Only in [35] the issue of dynamicity is considered among the requirements, although the solution proposed cannot be generalised. In our work we will focus on combining the need for information integration with the assumption that information changes dynamically, and with the fact that we address a large scale distributed system.

2.4. Data Grid and Data Grid Managment System (DGMS)

A Data Grid enables coordinated sharing of heterogeneous distributed storage resources and digital entities based on local and global policies across administrative domains in virtual organisations. The Data Grid Management System (DGMS) consists of a set of protocols and a hierarchical framework for coordinated datagrid management across administrative domains. The protocols or services collectively operate on groups of inter-organisational data using the behaviour specified by the data.

A Data Grid broker [36,37] act as an agent for an administrative domain in a DGMS framework. It facilitates sharing of services and data as components of active datagrid collections in the datagrid. In fact, both the Data Grid Management System and a Data Grid broker require information services (knowledge space manager, metadata catalogue service, and datagrid meta index, etc.) supporting the operations on data.

Compared with Data Grid Management System (DGMS), the ActOn-based information service focuses on providing "meaningful" information about whole Grid system, including distributed entities, their behaviour, and relationships among them. And DGMS is a data-oriented management system, which is for data sharing of Data Grid in a efficient way. However, applications (e.g., workflow management system) may need more information that DGMS cannot provide, such as a parallel computing environment, job scheduler, etc. One noticeable fact is that ActOn can integrate all necessary information for an application service based on the domain knowledge.

One interested work in [38] discussed the key concepts of Data Grid, and provided comprehensive taxonomies for describing Data Grids. Our ActOn Grid ontology go one step further to build Grid knowledge (instances, and the relationships among instances). By the ontology, the information will be able machine-automatic processing, and a reasoning engine or a rule engine can be used for manipulating on those semantic metadata for various purpose. Another interested work OGSA-DAI is for query heterogenous information sources. It focuses more on database query and query planning, however, does not provide semantic information integration.

3. The ActOn information integration approach

As described in the introduction, ActOn (Active Ontology) is an ontology-based information integration approach that can be used to generate and maintain up-to-date metadata for a dynamic, large-scale distributed system. In this section we will describe the main characteristics of this approach and its architecture, and will use as a running example the details of the EGEE information service that we have built with this approach.

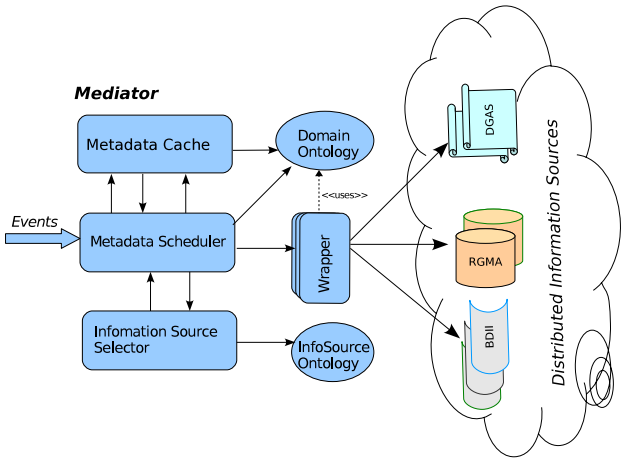


Fig. 3. Overview of the active ontology architecture.

3.1. Requirements for ActOn

The development of ActOn was based on a list of requirements that is based on the actual information integration needs that were identified in dynamic, distributed systems, such as the EGEE Grid [3], Crossgrid [39], Unicore [40], etc.

- We need to deal with frequent changes of metadata, caused by the dynamic features of the entities of a large-scale distributed system, in an efficient way, so that we avoid unnecessary continuous update information and we only change those parts that actually need changing.
- We need to be able to select the most suitable information source from a set of geographically-distributed and heterogeneous ones, which provide overlapping pieces of information, in different formats, and which can be available or unavailable at a given point in time.

Although these requirements arise in the context of developing integrated information services for Grid infrastructures, similar requirements can be also found in other application domains (e.g., the stock market, currency exchange, etc.). Therefore, ActOn provides a generic solution that can be easily adapted to different application domains.

3.2. The ActOn architecture

ActOn is comprised of a set of knowledge components, which represents knowledge from the application domain and from the information sources; and software components, such as a metadata scheduler (MSch), an Information Source Selector (ISS), a Metadata Cache (MC), and a set of Information Wrappers. Fig. 3 shows how these components are interrelated and how they are related to the corresponding information sources where data is taken from.

In the following sections we will describe all these components from the architecture, together with examples of the information service that we have built for EGEE, which illustrates their role and functioning in the system.

3.3. ActOn knowledge components: The ActOn information model

The knowledge components used in ActOn include a set of domain ontologies and an ontology about information sources, which are linked together. Domain ontologies (*DO*) describe the metadata information model in the form of domain concepts and properties for which instances will be generated, and restrictions about them. In the context of our EGEE information service these are resources, components, services, and applications of the EGEE Grid. The Information Source Ontology (*ISO*) provides information about the characteristics of information sources, which are used for the information source selection process. In our service they describe information services deployed in EGEE. The two ontologies are related by means of mappings (*Linker*) that specify which domain concepts and which of their properties can be generated by which information sources, as we will explain later.

3.3.1. Domain ontologies

Domain ontologies define the global information model used to represent metadata. In ActOn, they are used to represent and capture the configuration and state of the distributed system, representing Grid entities and their relationships as ontology classes, relations and individuals, as described in [41,42], for example. Different distributed systems may have different ontologies, which can be loaded into ActOn.

Although ActOn does not put any constraint about the ontology language to be used to implement these ontologies, in our implementation we use OWL [43] for this purpose.

3.3.2. Information Source Ontology

The Information Source Ontology defines the classes and properties of *information sources*, that is, network-enabled entities providing information about the configuration and state of DO elements. This ontology assists in locating suitable information sources for a specific information need. It describes the features of the information sources to be used by the system and is divided into a domain-independent part, with five classes and forty properties, and a domain-specific part that contains descriptions of the types of information sources that can be used in an application, as well as specific instances of these classes.

The most important class in the domain-independent part of the ontology is `InfoSrc`, which is described with four properties:

(i) `accessAPI`: it defines the information model and the information access methods to be used. For instance, the information model of BDII is LDAP, and its accessAPI can be "ldapsearch" in C and "JNDI" in Java;
(ii) `accessPoint`: it defines the server and port names to be used to obtain the information from. For instance, the CERN BDDII server can be described as "ldap://prod-bdii.cern.ch:2170";
(iii) `belongToMiddleware`: it specifies the middleware infrastructure (e.g., EGEE) where the information service is available, since depending on the middleware type and release being used the information access methods will be different;
(iv) `withSchema`: it indicates the kind of information that an information source provides. For instance, the EGEE BDII servers use the Glue Schema.

The domain-dependent part for our service contains descriptions of the following four main EGEE information providers: BDII (with the class `BDIIIP` being used to represent distributed BDII servers), RGMA, GridICE, and Unix-scripts. All of them are subclasses of the class `InformationSource`. Besides, we have defined 36 instances of BDIIIP, 10 instances of RGMA, 5 GridICE, and 10 Unix-script.

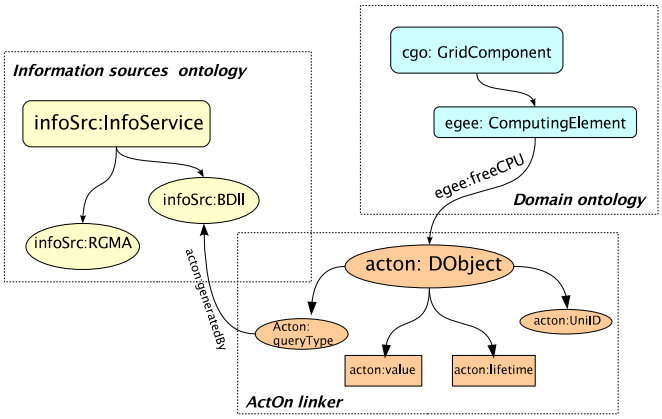


Fig. 4. The ActOn Semantic Model: Domain & Information Source Ontologies and the ActOn Linker.

An example of the information contained in one of the BDIIIP instances is:

* server name: ldap://prod-bdii.cern.ch
* server port: 2170
* access API: BDIIRet.class
* information schema: glueschema
* grid middleware: gLite middleware

3.3.3. Linker

As shown in Fig. 4, the association between the domain and information source ontologies is expressed by means of the *Linker*. Each domain ontology class or property is connected to the `DObject` class. The property `generatedBy` represents the means to be used to extract information from the source and transform it into the domain ontology components. Each of the mappings specifies, as well, the timestamp and lifetime of the information retrieved from the information sources. This information is used by the Metadata Scheduler to control the Metadata Cache, as explained later.

That is, ActOn allows the dynamic parts of an OWL class definition to be described by embedded queries, in a way similar to the AXML proposal [44]. Unlike the traditional assignment of fixed datatype values to properties of OWL instances, ActOn can assign query objects to time-variant properties. The query is embedded into an OWL class/instance definition, so that it can be dynamically executed. Hence, the values of the time-sensitive properties can be fetched dynamically in order to update the instances according to the changes that take place.

In principle, these queries can be presented in any kind of query language even as just a wrapper with a simple UNIX script. This makes ActOn more flexible and rigorous for maintaining the information about dynamic distributed systems.

3.4. ActOn software components

We will now describe the software components that comprise the ActOn architecture, as shown in Fig. 3.

3.4.1. Metadata scheduler (MSch)

It is designed to apply an update-on-demand policy to cache metadata. That is, the cached metadata is not updated until it is stale when being queried, so as to avoid unnecessary updates. We adopt event-driven mechanisms to cope with that policy. We have defined three types of event that can trigger the update process, though we have only implemented the first one in our service. They are:

(i) Application-specific events. They are application-based lifetime control events. The MSch can force an update process based on specific application requirements. For instance, an external application may require to update a specific piece of metadata at a given point in time.
(ii) Query events. They are raised when metadata is being queried. As we will show below, if the metadata being queried is available in the Metadata Cache and valid, the information sources are not contacted. If not, then we contact them to get fresh metadata.[1]
(iii) System-related events. They can cause changes of the Grid entities that the metadata refers to. A typical example is a *job-finished event*, which can cause the change of the value of the `runningJob` property of an instance of the class `JobQueue`.

The MSch acts upon receiving events. When the metadata scheduler receives a query event that involves retrieving metadata that has never been retrieved before or that is not valid since its expiry time has passed, or when it receives any of the other types of events, the metadata scheduler follows three steps: (1) it contacts the Information Source Selector to select the most suitable information source where to obtain the metadata from; (2) it retrieves the metadata from the selected sources, using the corresponding wrappers; and (3) it updates the Metadata Cache, assigns a time-stamp to the retrieved information and sends back the results to the requester.

An example can illustrate a typical procedure of MSch workflow (Fig. 6). When a query event is triggered that requests metadata for the Computing Element ce101.cern.ch, the MSch will first check the time-stamp of its associated metadata, which is stored by the Metadata Cache, and compare it with its lifetime. If it is valid, then it will just give back the results. If it is out of date, then it will invoke the Information Source Selector service to select a suitable information source (i.e., one EGEE region or site BDII server) for updating the Computing Element metadata. After getting the information about a suitable information source (for example, lxb2086.cern.ch or prod-bdii.cern.ch), it invokes the corresponding Information Wrapper service to fetch the information with an ldapsearch query, and then invokes the Metadata Cache to update (refresh) the metadata by modifying the values and time-stamp of the relevant properties. At the same time the new metadata is sent back to the metadata requestor.

Our approach has clear advantages over others that update metadata on a regular time-scale basis, such as Globus MDS and gLite BDII. These systems keep updating all their metadata every 6–8 min. This approach is too expensive and imprecise, particularly in large-scale distributed systems. On the one hand, there are many useless updates: a lot of updated metadata is most likely not being used (queried) in hours although it is updated every few minutes. On the other hand, some of the metadata may not be accurate in the case that the values of the metadata change more frequently than the regular update time. In fact, some of the dynamic metadata of BDII, such as freeCPU number, runningJobs or networking bandwidth, is usually incorrect as it is never updated on time.

3.4.2. Information Source Selector (ISS)

The Information Source Selector (ISS) is used to find the most suitable information source from the set of available sources, which are described as instances of the Information Source Ontology. Information sources can be any system (database, file, service, etc.) that contains relevant information. In Grid systems there are many redundant and geographically-distributed information sources available. For example, over 20 region BDII servers can be used to fetch information about the EGEE Computing Elements.

The selection is based on a set of retrieval conditions, including the actual information needed (specified as a SPARQL query), and other aspects like the geographical proximity of the source. For example, in our prototype we have defined the class `ComputingElement` that represents EGEE computing elements. This class has a property `freeCPU` that is `generatedBy` the information source BDII.

Since in our ontology we have defined over 30 BDII servers (as instances of the class `BDIIIP`), the ISS service sends a query to select the most suitable one for fetching the needed value. The query is done in SPARQL, and retrieves those instances of `BDIIIP` that `belongToMiddleware` EGEE Grid, whose `schema` is GlueSchema and whose `version` is 3.0. Also the middleware is *gLite*, and the release version *3.1.5*. Below is a SPARQL query for a BDIIIP instance in our implementation:

```
PREFIX  onG: <http://www.cs.man.ac.uk/img/ontogrid/>
FROM    <EGEEGridInfo.v0.3.owl>
SELECT  ?BDIIIP
WHERE   { ?x onG:runningService bdiiip?   .
          OPTIONAL { ?x onG:belongTo ''EGEE"  .
                     ?y onG:installedOn  ''gLite''  .
                     ?z onG:withSchema   ''GlueSchema'' . }
```

The selected BDIIIP instances are ranked according to their geographical proximity, quality of the service (QoS) and the capabilities of the BDII server machine. The ranking can be a dynamic activity. For example, QoS is based on the results of some BDII test scripts (e.g., ldapsearch - x -h ⟨hostname⟩-p 2170 -b "o = grid"). In my implementation, I define the "Good" status is TRUE only if there is a test success six times continuously; and I define the "Bad" status is TRUE whenever there is one time the test failed.

3.4.3. Information Wrappers

After an information source is selected, the Metadata Scheduler contacts the corresponding Information Wrapper in order to retrieve the relevant up-to-date information. Normally there is an Information Wrapper per type of information source accessed (that is, one for MDS, another one for BDDII, etc.). We have developed four kinds of wrappers: the BDII server wrapper, the RGMA server wrapper, the GridICE wrapper, and the Unix-script wrapper.

The wrappers are used to fetch information from different information sources. First, the Information Wrapper gets information from the information source ontology about the data model of the specific source to be accessed, and about its access API and access point. Then it fetches the information from its source. For instance, a BDIIIP information source can be queried using an LDAP query based on the information from a BDII individual, such as "ldapsearch -x -H ldap://prod-bdii.cern.ch:2170-b mds-vo-name = CERN-PROD, o = grid". Once the query is answered, the results are transformed into instances of the concept `ComputingElement` of the domain ontology.

ActOn does not impose any specific technology for generating Information Wrappers. They can be generated in an ad-hoc manner, by hard-coding the access to the information source and the transformation into the application domain ontology. They can be also generated with generic wrapper-generation languages and technologies, such as WSL [45], D2R [46], R2O [47], etc.

3.4.4. Metadata Cache (MC)

The Metadata Cache (MC) stores and manages the metadata obtained from the information sources, together with its timestamp and lifetime information, so that it can check whether such property values are still valid or not (e.g., lifetime control) when it receives a query event that involves them.

[1] In the case that the latency is bigger than the update time of the information source, this will still provide out-of-date metadata, but in the rest of cases data will be always up-to-date.

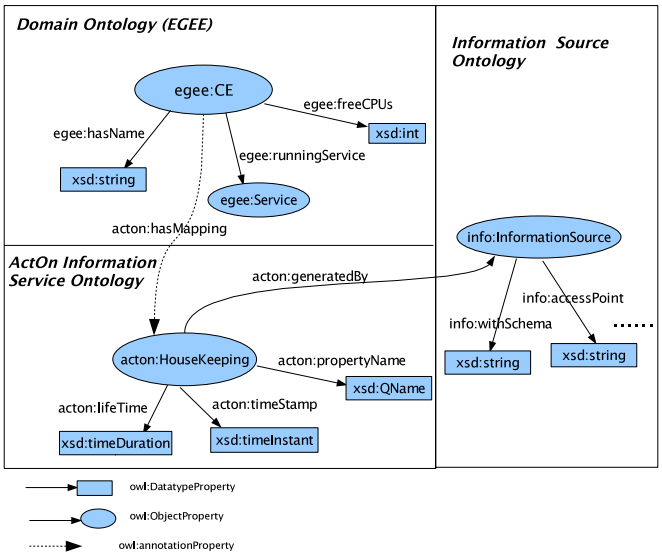


Fig. 5. Graphical overview of the association between domain and information source ontologies.

The Metadata Cache uses the domain ontologies as its information model. For instance, in our service the MC caches information about Computing Elements (CE), Storage Elements (SE), Virtual Organisations (VO), etc. As commented above, the MC uses the S-OGSA semantic binding service implementation in order to store the values together with their timestamp and lifetime, using the mappings shown in Fig. 5. The information stored in MC can be queried using the SPARQL query language [48].

4. Prototype implementation

The ActOn semantic information service is part of S-OGSA (Semantic-Open Grid Service Architecture) middleware service [27], and hence it is Web Service Resource Framework (WS-RF) [49] compliant.

The ActOn information service work together with S-OGSA Semantic Binding Service (SBS). The SBS is used to bind semantic metadata with the ontologies it refers to and with the resources that the metadata describes, so that metadata can be managed as a resource, with its own lifetime, authorisation policies, etc. All the source code of ActOn and of the information service that we have described is available under Open Source license at the OntoGrid CVS [50].

5. Information quality evaluation and comparison

In this section we show how we have designed and run the experiments to evaluate our ActOn-based EGEE information service, and we discuss the results obtained, in comparison with other information services deployed and available in the EGEE infrastructure (BDII and RGMA[2]).

In the following sections we will describe our evaluation framework for information quality, including the design rationale, the experiments to be carried out, and the metrics to be used for the evaluation, together with details about how they are measured for each system.

5.1. An evaluation framework for information quality in grids

Information quality (IQ) can be defined as a measure of the value of the information provided by an information system to its users [51]. There are many characterisations of what quality means in this context (taking into account that quality is normally subjective and depends on the intended use of the information by users). The authors in [51] distinguish between intrinsic, contextual, representational and accessibility IQ, and define different factors to be considered for each of them (accuracy, objectivity, reputation, relevancy, etc.).

[52,53] propose to focus on seven of these characteristics, which are considered the most important ones, independently of their domain: completeness, accuracy, provenance, conformance to expectations, logical consistency and coherence, timeliness, and accessibility. In our framework we have selected three of these features, namely *completeness*, *accuracy* and *conformance to expectations*.

We are not worried about the *provenance* of information, since we know clearly which are the information sources that we use in each moment and which are the information providers responsible for that information. We are not worried either about *accessibility*, since we assume that the systems work within a Grid security infrastructure (e.g., GSI [54]), so that the information is accessible as long as the client has the corresponding rights to access it and knows the information model and API used by the corresponding information service.

With respect to the *logical consistency and coherence* and the *timeliness* of the information retrieved and aggregated from the information sources, these are features that will form part of our future evaluation work, and will be also considered in further developments of the ActOn-based information service. An example of why the first feature is important is the following: there are many cases where a computing element specifies that it gives support to MPI but does not comply with the requirements for running an MPI job, which are that it must be a CE server, must have an `sshd` service running on it, must have the libraries `mpirun`

[2] MDS is not included in our evaluation because it is not deployed as an information service of the EGEE Grid. However, the results obtained for BDII can be easily translated into MDS, since BDII is an extension of that information service.

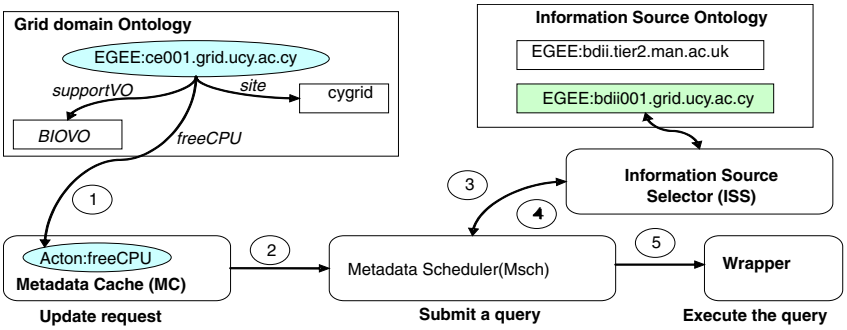


Fig. 6. A typical procedure of MSch workflow.

and `libmpi.so` in its file system, and must have at least two worker nodes. Information services like BDII or RGMA only store and provide the information that their information producers give them, without checking their consistency, hence they provide incorrect information due to this fact. As an example of the second feature, BDII normally updates the information that has been provided by its information sources every five or six minutes, what means that this information may be already inaccurate when a client requests it. Hence, having metadata about the lifetime and freshness of information in the information service is important.

5.2. Experiment metrics

To check the three criteria considered in our framework, we are interested in knowing whether all information services obtain the same results when answering the same query, given the same conditions in the EGEE testbed. We also want to check how many of those answers are correct and how many of the existing answers are actually retrieved. This also allows us to know whether the results provided by the services conform to the expectations of the users. To check this, we have selected two metrics, commonly used in information retrieval: precision and recall. Below we provide their definitions and the formulae used to calculate them:

Precision: The proportion of relevant information retrieved, out of all the information retrieved.

$$\text{Precision} = \frac{(\textit{relevant information}) \cap (\textit{retrieved information})}{\textit{retrieved information}}. \quad (1)$$

Recall: The proportion of relevant information that is retrieved, out of all the relevant information available.

$$\text{Recall} = \frac{(\textit{relevant information}) \cap (\textit{retrieved information})}{\textit{relevant information}}. \quad (2)$$

5.3. Experiment setup and design

We have designed a set of experiments for measuring the information quality criteria selected. Measurements are taken on a real Grid testbed, the EGEE production testbed, which at the time of the experiments, has gLite 3.0.1 installed as its middleware. The user interfaces used to access the EGEE Grid are the UI machines at the University of Manchester,[3] United Kingdom, and at the Institute of Physics of Belgrade,[4] Serbia.

To carry out the experiments and record their results, we have developed a set of Java-based client software and Unix shell scripts.[5]

The key aspects upon which we compare different information services are: (i) the information model that each information service adopts; and (ii) the expressiveness of its query language. In order to evaluate these two features, we have proposed six common queries that are frequently issued by middleware systems like schedulers, resource brokers or by more complex applications:

- Query 1: Find all the Computing Elements (CEs) that support the BIOMED Virtual Organisation (VO).
- Query 2: Find all the CEs that support the BIOMED VO and have more than 100 CPUs available.
- Query 3: Find all the CEs that support the MPI running environment.
- Query 4: Find all the CEs that support the BIOMED VO, have more than 100 CPUs available, and support the MPI running environment.
- Query 5: Find all the CEs where GATE (Geant4 Application for Tomographic Emission) can be run.
- Query 6: Find all the CEs that support the BIOMED VO, have more than 100 CPUs available, and where GATE can be run.

Table 2
An Example of the Query 1 in BDII, RGMA, and ActOn.

Infor. Service	Query 1
BDII (LDAP Search)	`ldapsearch -x -H ldap://lcg-bdii.cern.ch:2170 -b mds-vo-name=local,o=grid ' (&(objectClass=GlueVOView) (GlueVOViewLocalID=biomed))' GlueCEAccessControlBaseRule`
RGMA (SQL Query)	`select GlueCEVOViewUniqueID, Value from GlueCEVOViewAccessControlBaseRule WHERE Value='VO:biomed'`
ActOn (SPARQL Query)	`PREFIX egeeOnto: <http://www.cs.man.ac.uk/img/ontogrid#> SELECT ?ceid ?ceID ?VO WHERE ?ceid egeeOnto:CEUniqueID ?ceID. ?ceid egeeOnto:hasVO ?VO. OPTIONAL ?ceid egeeOnto:VO ?ceID. FILTER (?vo = ''biomed'')`

Each of these six queries has been translated into the query languages of the three information services. Table 2 shows an example for Query 1. And we use different query client tools to execute these queries and extract the results obtained (e.g., ldapsearch for BDII, the gLite RGMA client tools for RGMA and a Java-based ActOn client for the ActOn-based information service.

Not only are queries different, but also query results are obtained in different manners, due to the differences in the information models of each service. The result of a BDII query is a set of LDAP entries, of an RGMA query a set of table rows, and of an ActOn-based query a set of RDF triples. Fig. 7 shows three different ways to show the same Grid resource in the three services evaluated (i.e., ce02.tier2.hep.manchester.ac.uk, an EGEE Computing Element). Even if they have different syntax and size, in our experiment we count them as one piece of information each. That is, we use each "Grid resource" obtained from a query as the basic unit for counting information, which will be used to calculate precision and recall, as described below.

The experiment consists of examining the information retrieved for each of the six queries aforementioned, so as to get their corresponding precision and recall measures.

Precision is easy to determine, since it can be computed manually by looking at the results obtained from each query. In all cases, we assume binary relevancy of information, that is, each piece of information retrieved is either relevant or irrelevant for the issued query.

Recall is more difficult to determine, due to the fact that the amount of information available in the EGEE production testbed changes frequently in these systems and there is no way to get accurate information about the actual state of the Grid resources that are available without using the information services that we are evaluating. To get a good approximation that can be used for our purposes, we execute each query 100 times, with a 4-min interval between executions, that is, we monitor the testbed during 400 min. Then we use the highest value obtained from these 100 executions as the total amount of relevant information to be used to calculate recall.

Tables 3–5 provide the precision and recall measurements obtained after the execution of the experiments described above for the three information services selected: BDII, RGMA and the ActOn-based information service. The values provided in the tables show the average of executing the queries 100 times.

[3] ui.tier2.hep.manchester.ac.uk.

[4] ce.phy.bg.ac.yu.

[5] They are available in the IST OntoGrid project CVS repository [55].

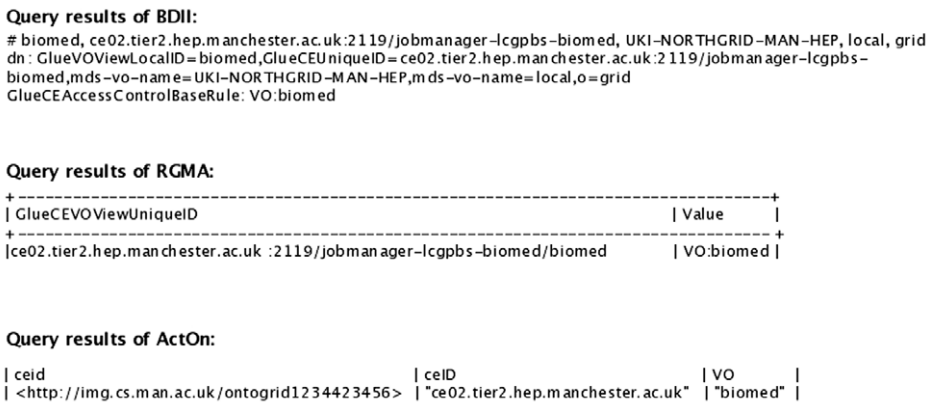


Fig. 7. Results of BDII, RGMA, and ActOn for the the same Grid resource Computing Element at University of Manchester (ce02.manchester.ac.uk).

Table 3
BDII Recall & Precision Measurement (100 times).

Query No.	Retrieved Info.	Relevant Info.	Precision	Recall
1	14 999	15 200	1	0.987
2	242 517	19 708	0.082	0.918
3	7 174	7 300	1	0.983
4	485 034	4 600	0.010	0.990
5	–	–	–	–
6	–	–	–	–

Table 4
RGMA Recall & Precision Measurement (100 times)

Query No.	Retrieved Info.	Relevant Info.	Precision	Recall
1	3 417	15 200	1	0.225
2	6 321	6 321	1	1
3	6 568	7 300	1	0.900
4	11 245	4 914	0.437	0.563
5	–	–	–	–
6	–	–	–	–

Table 5
ActOn Recall & Precision Measurement (100 times).

Query No.	Retrieved Info.	Relevant Info.	Precision	Recall
1	15 200	15 200	1	1
2	34 100	34 100	1	1
3	6 568	7 300	1	0.900
4	6 568	7 300	1	0.900
5	24	24	1	0.900
6	6	6	1	1

As a general comment about these results, we can highlight the fact that BDII shows in general poor results with respect to recall and precision, while ActOn and RGMA present better results. This is mainly related to the repository that BDII uses (LDAP), which is too lightweight and hence provides weak information process and query capabilities; while RGMA's is based on relational databases and ActOn's is based on RDF, which both have better query capabilities.

Now we will analyse with more detail some of the system behaviours over specific queries, and derive more conclusions from these values:

- *BDII has weak query capabilities*. Table 3 shows that BDII has extremely bad precision results for queries 2 and 4, while the results for queries 1 and 3 are excellent. This is related to its weak query ability, as aforementioned. LDAP-based queries are string-based, and hence they cannot be used to support queries over numerical values, such as "greater than or lower than". If we want to improve this precision value, we need to fetch all the information about CE CPUs as a string value first (as we have done to get these results), and then post-process (filter) those results on the client side. RGMA and the ActOn-based information services do not have that problem, since their query abilities are better.
- *RGMA is not able to relate information available in different tables*. Table 4 shows that RGMA has bad precision results in query 4. RGMA contains information to solve this query, but the information comes from two different tables (`GlueCE` and `GlueSubClusterSoftwareRunTimeEnvironment`), and the query language used by RGMA does not allow making a join of both tables. Hence the situation is similar to the previous case: this problem can be solved on the client side by post-processing the results that have been obtained from each separate query.
- *RGMA is very sensitive to the registering and availability of information providers at a given point in time*. Table 4 shows that RGMA has bad recall results in query 1. This is because numbers of Computing Element producers that are available during the experiment are not always stable, due to the fact that either producers were not registered in the RGMA registry at that specific moment, or that the producers were not configured correctly or available at that point in time. BDII and the ActOn-based information service are more robust to this, due to the fact that they store information locally and do not depend on their information providers at the time of querying.
- *Some complex queries cannot be answered by one type of information service in isolation*. Tables 3 and 4 show that BDII and RGMA can only answer the first four queries. They cannot answer queries 5 and 6 because their information providers cannot provide enough information and should be combined. This shows that the ability of BDII and RGMA to share their data resources is weak. On the other hand, the ActOn-based information service has the ability to adopt existing information sources as its information providers, and aggregate information from these information sources to answer such complex queries.

Finally, Fig. 8 shows the response time of the different information services for the queries identified in our information quality experiments.[6] We do not include a deep discussion about these experiments since performance is out of the scope of this paper. However, we want to show that there is a reasonable trade-off between information quality and response time of our solution. The main summary that can be obtained from these results is that BDII and the ActOn-based information service are similar with respect

[6] The data in which this graphical representation is based is available at [50].

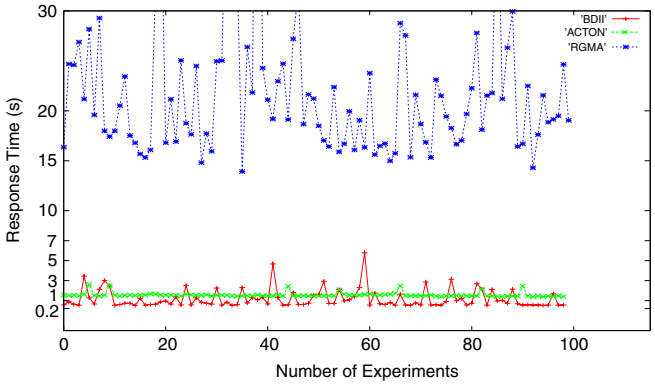


Fig. 8. Query Reponse Time of BDII, RGMA, and ActOn for the the same Grid resource Computing Element at University of Manchester (ce02.manchester.ac.uk).

to their response time, since both of them have caches, and RGMA is generally slower than them, due to its information management architecture.

5.4. Lessons learned

The experience of developing the experiments for information quality measurement and conducting them on the EGEE Grid testbed has generated several valuable lessons, most of them related to the fairness of the information quality measurement process, which can be applicable to other similar types of experiment.

First, it is difficult to find standard domain-independent methods to measure information quality in information systems. Hence if we want to design and run an experiment in a specific domain (e.g., Grid information services), we must design it according to that domain and the information needs of the information service users (either other applications or end-users).

Second, different information services use different information models, and usually provide different expressivity in their query languages or access APIs. This means that a special effort has to be made in order to define clearly a fair way to perform measurements that takes into account these differences.

Third, the proposed centralised architecture provides good support to the performance requirements needed in such a dynamic environment. However, this does not mean that this architecture could be similarly applied in a more decentralised nature, using several ActOn information systems organised as a P2P system for information sharing among Grids.

We think that the results that we have presented can be of great help for the developers who work in the implementation of these and other similar information services, so that they can use these experiments as a benchmark suite, and for the developers of information-intensive applications that make use of these services.

6. Conclusions and future work

In this paper we have presented an ontology-based information integration approach, Active Ontology (ActOn), which overcomes some of the limitations of current similar approaches when dealing with highly dynamic, distributed and redundant information sources in the cases where information quality, availability and robustness, as well as response time, are important non-functional requirements.

We adopt a data warehouse approach to information integration, where we materialise relevant information from different information sources and assign it a lifetime based on the update frequency of the information sources where it is taken from. The materialised information acts as a Metadata Cache that is updated only when an information request is sent to the system and the materialised information has expired.

Besides, information sources are selected at run-time from a large set of sources that provide redundant information, based on criteria such as their information coverage, availability, geographical proximity, etc.

The results of the experiments executed to analyse the quality of metadata and the response time of our system are promising, suggesting that it can increase the metadata quality and robustness of currently-deployed information systems, and decrease the cost of system resources.

In summary, our main contribution over the state of the art in Grid information systems is that we have proposed a Grid information service that performs an ontology-based integration of information from existing services, what allows automatically creating execution plans for retrieving information from sources that are overlapping in the information that they publish and have different provenance constraints, and maintain a cache of relevant information as long as it is valid given its lifetime constraints.

As for the integration of features from other systems, we plan to work on the integration and extension of (semi-)automatic wrapper generation systems like D2R and R2O (currently these systems are only available to access databases, but we plan to extend them for accessing information services such as those present in Grid systems), and on the integration of query reformulation and planning techniques, such as those of Theseus [19], with the Metadata Cache approach that we have proposed.

We also plan to take full advantage of following an ontology-based approach for information integration, allowing us to perform tasks that cannot be done easily with the services currently available, such as detecting inconsistencies in the metadata that is available or deriving new information. For example, a common problem with current information services is their level of trustiness. There are many cases where a computing element specifies that it gives support to MPI but does not comply with the requirements for running an MPI job, which are that it must be a CE server, must have an `sshd` service running on it, must have the libraries `mpirun` and `libmpi.so` in its file system, and must have at least two worker nodes. Similarly, we could derive that a computing element gives support to MPI if the previous conditions apply, since this is a necessary and sufficient condition.

Finally, we will explore other usage scenarios with similar non-functional requirements, in terms of highly dynamic, distributed and possibly redundant sources, such as the stock market or the currency exchange domains.

Acknowledgements

This work is supported by the EU FP6 OntoGrid project (STREP 511513) funded under the Grid-based Systems for solving complex problems, and by the Marie Curie reintegration grant WS-DAIOnt-OWL (FP6-2004-Mobility-11-046415). We also thank CoreGRID Network of Excellence (Contract Number 004265) and the other members of the OntoGrid team at Manchester for their helpful discussions, as well as members of the IMG group (Ian Horrocks and Rizos Sakellariou), and of EGEE (Antun Balaz and Laurence Field).

References

[1] I. Foster, C. Kesselman, J. Nick, S. Tuecke, The physiology of the grid: An open grid services architecture for distributed systems integration, in: Open Grid Service Infrastructure WG, Global Grid Forum, 2002.
[2] I. Foster, C. Kesselman, The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 1999.
[3] Enabling Grids for E-sciencE, EGEE. http://public.eu-egee.org/.
[4] Berkeley Database Information Index, BDII. http://lfield.home.cern.ch/lfield/cgi-bin/wiki.cgi?area=bdiipage=documentation.

[5] K. Czajkowski, S. Fitzgerald, I. Foster, C. Kesselman, Grid information services for distributed resource sharing, in: Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing, HPDC-10, IEEE Press, 2001.
[6] EDG RGMA. www.marianne.in2p3.fr/datagrid/documentation/rgma-guide.pdf.
[7] J.M. Schopf, M. D'Arcy, N. Miller, L. Pearlman, I. Foster, C. Kesselman, Monitoring and discovery in a web services framework: Functionality and Performance of Globus Toolkit MDS4, Argonne National Laboratory, Tech. Rep., 2005.
[8] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, S. Hübner, Ontology-based integration of information — A survey of existing approaches, in: H. Stuckenschmidt (Ed.), IJCAI–01 Workshop: Ontologies and Information Sharing, 2001, pp. 108–117.
[9] W. Xing, O. Corcho, C. Goble, M.D. Dikaiakos, Acton: A semantic information service for egee, in: Proceedings of the 8th IEEE/ACM International Conference on Grid Computing, Grid 2007, pp. 81–88.
[10] L. Bellatreche, D.N. Xuan, G. Pierra, H. Dehainsala, Contribution of ontology-based data modeling to automatic integration of electronic catalogues within engineering databases, Computers in Industry Journal (2006) 711–724.
[11] J.D. Ullman, Information integration using logical views, in: ICDT'97: Proceedings of the 6th International Conference on Database Theory, Springer-Verlag, London, UK, 1997, pp. 19–40.
[12] F.G.V. Lattes, M.-C. Rousset, The use of CARIN language and algorithms for information integration: The PICSEL system, International Journal of Cooperative Information Systems 9 (4) (2000) 383–401.
[13] C. Reynaud, G. Giraldo, An application of the mediator approach to services over the Web, in: J. Cha, R. Jardin-Gonçalvez, A. Steiger-Garçao (Eds.), in: Concurrent Engineering, vol. 1, A.A. Balkema Publishers, 2003, pp. 209–216.
[14] A.Y. Levy, A. Rajaraman, J.J. Ordille, The world wide web as a collection of views: Query processing in the information manifold, in: Proceedings of the International Workshop on Materialized Views: Techniques and Applications, VIEWS 1996, 1996, pp. 43–55.
[15] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J.D. Ullman, J. Widom, The TSIMMIS project: Integration of heterogeneous information sources, in: 16th Meeting of the Information Processing Society of Japan, Tokyo, Japan, 1994, pp. 7–18.
[16] M. Michalowski, J.-L. Ambite-Molina, S. Thakkar, R. Tuchinda, C. Knoblock, S. Minton, Retrieving and semantically integrating heterogeneous data from the web, IEEE Intelligent Systems 19 (3) (2004) 72–79.
[17] Y. Arens, C.A. Knoblock, W.-M. Shen, Query reformulation for dynamic information integration, Journal of Intelligent Information Systems 6 (2–3) (1996) 99–130.
[18] L. Padgham, M. Winikoff, Prometheus: A methodology for developing intelligent agents, in: Proceedings of the Third International Workshop on Agent-Oriented Software Engineering, AAMAS 2002, Bologna, Italy, July 2002.
[19] G. Barish, D. DiPasquo, C.A. Knoblock, S. Minton, Dataflow plan execution for software agents, in: C. Sierra, M. Gini, J.S. Rosenschein (Eds.), Proceedings of the Fourth International Conference on Autonomous Agents, ACM Press, Barcelona, Spain, 2000, pp. 138–139.
[20] M.P. Singh, P. Cannata, M.N. Huhns, N. Jacobs, T. Ksiezyk, K. Ong, A.P. Sheth, C. Tomlinson, D. Woelk, The carnot heterogeneous database project: Implemented applications, Distributed and Parallel Databases 5 (2) (1997) 207–225.
[21] R.J. Bayardo Jr., W. Bohrer, R. Brice, A. Cichocki, J. Fowler, A. Helal, V. Kashyap, T. Ksiezyk, G. Martin, M. Nodine, M. Rashid, M. Rusinkiewicz, R. Shea, C. Unnikrishnan, A. Unruh, D. Woelk, InfoSleuth: Agent-based semantic integration of information in open and dynamic environments, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, vol. 26, 2, ACM Press, New York, 1997, pp. 195–206. 13–15.
[22] E. Mena, A. Illarramendi, V. Kashyap, A. Sheth, OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies, International Journal on Distributed And Parallel Databases (DAPD) 8 (2) (2000) 223–272.
[23] C. Goble, R. Stevens, G. Ng, S. Bechhofer, N. Paton, P. Baker, M. Peim, A. Brass, Transparent access to multiple bioinformatics information sources, IBM Systems Journal 40 (2) (2001) 534–551.
[24] M.A. Jeusfeld, C. Quix, M. Jarke, Design and analysis of quality information for data warehouses, in: International Conference on Conceptual Modeling/the Entity Relationship Approach, 1998, pp. 349–362.
[25] J. Contreras, V.R. Benjamins, M. Blázquez, S. Losada, R. Salla, J. Sevilla, D. Navarro, J. Casillas, A. Mompó, D. Patón, Ó. Corcho, P. Tena, I. Martos, A semantic portal for the international affairs sector, in: EKAW, 2004, pp. 203–215.
[26] T.G. Team, Gt 4.0: Information Services (MDS4). http://www.globus.org/toolkit/docs/4.0/info/.
[27] Ó. Corcho, P. Alper, I. Kotsiopoulos, P. Missier, S. Bechhofer, C.A. Goble, An overview of S-OGSA: A reference semantic grid architecture, Journal of Web Semantics 4 (2) (2006) 102–115.
[28] OntoGrid Project. http://www.ontogrid.net/.
[29] H. Zhuge, China's e-science Knowledge Grid environment, IEEE Intelligent Systems 19 (1) (2004) 13–17.
[30] H. Zhuge, Y. Xing, P. Shi, Resource space model, owl and database: Mapping and integration, ACM Transactions on Interet Technology 8 (4) (2008) 1–31.
[31] H. Zhuge, X. Sun, A virtual ring method for building small-world structured p2p overlays, IEEE Trans. on Knowl. and Data Eng. 20 (12) (2008) 1712–1725.
[32] Semantic Grid and Knowledge Grid the Next Generation Web, vol. 20, no. 1, pp. 1–178, 2004. [Online]. Available. http://www.sciencedirect.com/science/journal/0167739X.
[33] J. Shen, Y. Yang, Extending rdf in distributed knowledge-intensive applications, Future Generation Computer Systems 20 (1) (2004) 27–46.
[34] M. Li, P. van Santen, D.W. Walker, O.F. Rana, M.A. Baker, Sgrid: A service-oriented model for the Semantic Grid, Future Generation Computer Systems 20 (1) (2004) 7–18.
[35] A.V. Smirnov, M. Pashkin, N. Chilov, T. Levashova, Knowledge logistics in information grid environment, Future Generation Computer Systems 20 (1) (2004) 61–79.
[36] S. Venugopal, R. Buyya, L. Winton, A grid service broker for scheduling e-science applications on global data grids, Concurrency and Computation: Practice and Experience 18 (2006) 685–699.
[37] C. Baru, R. Moore, A. Rajasekar, M. Wan, The sdsc storage resource broker, in: CASCON'98: Proceedings of the 1998 Conference of the Centre for Advanced Studies on Collaborative research, IBM Press, 1998, p. 5.
[38] S. Venugopal, R. Buyya, K. Ramamohanarao, A taxonomy of Data Grids for distributed data sharing, management, and processing, ACM Computing Surveys 38 (2006).
[39] J. Marco, et al., First prototype of the crossgrid testbed, in: Proceedings of First European AcrossGrids Conference, AXGrids 2003, in: LNCS, vol. 2970, Springer-Verlag, Santiago de Compostela, Spain, 2003, pp. 67–77.
[40] P. Wieder, D. Mallmann, UniGrids - uniform interface to grid services, in: 7th HLRS Metacomputing and Grid Workshop, Stuttgart, Germany, April 2004.
[41] W. Xing, M.D. Dikaiakos, R. Sakellariou, A core grid ontology for the semantic grid, in: Proceedings of the 6th IEEE International Symposium on Cluster Computing and the Grid, CCGrid 2006, IEEE Computer Society, Singapore, 2006, pp. 178–184.
[42] M. Parkin, S. van den Burghe, O. Corcho, D. Snelling, J. Brooke, The knowledge of the grid: A grid ontology, in: Proceedings of the 6th Cracow Grid Workshop, Cracow, Poland, October 2006.
[43] P. Patel-Schneider, P. Hayes, I. Horrocks, OWL web ontology language semantics and abstract syntax, World Wide Web Consortium, February 2004.
[44] T.A.X. Team, Active XML Primer. http://activexml.net/.
[45] H. Garcia-Molina, Y. Papakonstantinou, A.R.a.D. Quass, Y. Sagiv, J. Ullman, V. Vassalos, J. Widom, The TSIMMIS approach to mediation: Data models and languages, Intelligent Information Systems 8 (2) (1997) 117–132.
[46] C. Bizer, D2R MAP: A DB to RDF mapping language, in: 12th International World Wide Web Conference, Budapest, May 2003.
[47] J. Barrasa, O. Corcho, A. Gomez-Perez, R2O, an extensible and semantically based database-to-ontology mapping language, in: Proceedings of the 2nd Workshop on Semantic Web and Databases, SWDB2004, Toronto, Canada, 2004.
[48] E. Prud'hommeaux, A. Seaborne, SPARQL query language for RDF, W3C Working Draft, July 2005.
[49] O. Consortium, Web services resource framework primer, May 2006. http://docs.oasis-open.org/wsrf/wsrf-primer-1.2-primer-cd-02.pdf.
[50] OntoGrid CVS. http://www.ontogrid.net/ontogrid/downloads.jsp.
[51] R. Wang, D. Strong, Beyond accuracy: What data quality means to data consumers, Management Information Systems 12 (4) (1996) 5–34.
[52] N. Dushay, D. Hillman, Analyzing metadata for effective use and re-use, in: The International DCMI Metadata Conference and Workshop, 2003.
[53] B. Hughes, Metadata quality evaluation: Experience from the open language archives community, in: ICADL, 2004, pp. 320–329.
[54] I. Foster, C. Kesselman, G. Tsudik, S. Tuecke, A security architecture for computational grids, in: 5th ACM Conference on Computer and Communications Security Conference, 1998.
[55] OntoGrid CVS at UoM. http://rpc314.cs.man.ac.uk.

Wei Xing is a research project manager at InforSense Ltd. London. Before that, he worked at Information Management Group, University of Manchester. From 2002 to 2006, he was research associate of High Performance Computing Lab (HPCL), Department of Computer Science at the University of Cyprus. Wei Xing has over 20 publications in journals, and refereed scientific conferences. His current research interests focus on workflow, Semantic Web Technologies, SOA.

Oscar Corcho is an assistant professor at Universidad Politcnica de Madrid (UPM). He graduated in Computer Science from UPM in 2000, and received the third Spanish award in computer science from the Spanish Government. He obtained his PhD in Artificial Intelligence in 2004. His research activities include Ontological Engineering, the Semantic Web and Grid. He has participated in a large number of European and international projects in these areas, and has published two books, over 50 journal and conference papers, and reviews papers in many conferences, workshops and journals. He has collaborated in the organisation of conferences like EKAW2002, ESWC2006 and ESWC2008.

Carole Goble (http://www.cs.man.ac.uk/~carole/) is Director of the myGrid project, co-Director of ESNW and IMG and PI on OMII-UK and its Director for the first year. She led the first phase of the TAMBIS ontology based life science integration platform. Her research interests are in ontologies, the Semantic Web and the Semantic Grid, and their application to e-Science. She was a founding leader of the UK's e-Science activity. She is Technical Director of the OntoGrid EU FP6 project, which has devised the first reference architecture for the Semantic Grid, and Research Director of the EU Knowledge Web NoE. She has published over 160 papers, has served on all the prestigious conferences in the various fields she embraces, and has given numerous keynote talks in top conferences. She currently serves on 17 international advisory committees in semantics, Grid and eScience. She co-founded Cerebra, a Semantic Web company, which was sold to Web Technologies in 2006.

Marios D. Dikaiakos is an Associate Professor of Computer Science at the University of Cyprus, where he established and leads the High-Performance Computing Systems Laboratory and serves as Vice Chairman of the Computer Science Department. He received his Ph.D. from Princeton University (1994). Dr Dikaiakos' research focuses on network-centric computing, with an emphasis on Grids, Vehicular Ad-Hoc Networks, and the World-Wide Web. Dr. Dikaiakos has been a principal investigator for several projects funded by the European Union and the Research Promotion Foundation of Cyprus.