Active Ontology: An Information Integration Approach for Dynamic Information Sources

Wei Xing[1], Oscar Corcho[1], Carole Goble[1], Marios D. Dikaiakos[2]
[1] **School of Computer Science, University of Manchester, UK**
[2] **Department of Computer Science, University of Cyprus, Cyprus**
Email:{wei.xing, oscar.corcho, carole.goble}@manchester.ac.uk, mdd@cs.ucy.ac.cy

Abstract. In this paper we describe an ontology-based information integration approach that is suitable for highly dynamic distributed information sources, such as those available in Grid systems. The main challenges addressed are: 1) information changes frequently and information requests have to be answered quickly in order to provide up-to-date information; and 2) the most suitable information sources have to be selected from a set of different distributed ones that can provide the information needed. To deal with the first challenge we use an information cache that works with an update-on-demand policy. To deal with the second we add an information source selection step to the usual architecture used for ontology-based information integration. To illustrate our approach, we have developed an information service that aggregates metadata available in hundreds of information services of the EGEE Grid infrastructure.

1 Introduction and Motivation

As Grids grow larger and gain widespread use, there is an increasing need for rich and meaningful information about heterogeneous, massive Grid entities and resources. For example, there are currently over 20,000 CPUs available, 5 Petabytes of storage space in hundreds of storage elements, and an average of 20,000 concurrent jobs in the EGEE production testbed, having information about those heterogeneous entities is critical for the EGEE gLite middleware [1]. This information is used for tasks such as resource discovery, workflow orchestration, meta-scheduling, and security. Such information is normally aggregated and provided by information services, which can be defined as "databases of attribute metadata about resources" [2][1] Examples of information services are BDII [3] and MDS [4], focused on hardware and software resources; and RGMA [5], focused on jobs, services and running environments.

The main limitation of existing information services is that they do not provide enough and accurate information about large-scale distributed systems like EGEE, since they only focus on a few specific aspects of such systems. Therefore, there is a need for information services that provide "*Just Enough, Just in Time*" metadata for large-scale Grid systems. One solution can be the provision of meta-aggregation services of all essential information sources, giving a single information access point to all aspects of a Grid system in an efficient and economic way.

To aggregate such distributed information we need to address several **challenges**, raised by the dynamic and heterogeneous nature of Grids:

[1] In the rest of the paper, we will use the terms information and metadata interchangeably.

- **Metadata about most of Grid entities needs to be updated very 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.
- **The latency of retrieving a piece of information may be longer than its change frequency**. This is normally due to the fact that information sources are usually distributed in a wide-area networking environment, hence the network latency can have a strong influence in the quality of the metadata that is generated. Many applications are not very sensitive to small value changes (e.g., schedulers usually work with approximate information about availability of the resources, such as number of CPUs available, job queue status, free storage space, network throughput, etc.), but in some cases applications or services may require precise metadata in order to deliver a good quality of service.
- **Metadata of a Grid entity consists of multiple attributes, whose values can be normally obtained from different geographically-distributed information sources**. In a large-scale Grid system there are usually several information sources that can provide the same piece of information about a resource. And it is difficult to identify and locate the most suitable and available information source for a specific information need.

Ontology-based information integration has been traditionally used to create aggregated information services that can be used to query multiple information sources transparently [6]. Among these 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). 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 4.

– **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 so frequently as it is the case in Grid systems. Namely, the information is assumed to be *valid* for a long time, that is, longer than what it is needed to execute the query and aggregate the information. Clearly, 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 time-scales. 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, the information about membership to a virtual organisation may change in days.
– **Ontology-based information integration systems are not fault-tolerant and robust**. 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 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 source, with different service quality and cost. Hence, robust fault-tolerant aggregation 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.

These limitations are addressed in our approach: Active Ontology (ActOn). ActOn 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**, which can be a Grid system or not. To achieve this, we reuse architectural ideas and techniques that have been proposed in the context of ontology-based information integration. We add an intermediate information source selection step that takes into account the current information needs and the state of the information services to be accessed. We also include a "metadata cache" that works with an update-on-demand policy, so that only currently used metadata is aggregated, without the need for continuous update requests. To demonstrate our approach, we show a prototype implementation for the aggregation of metadata coming from several information sources in the EGEE Grid.

The remaining of this paper is organised as follows. Section 2 presents the architecture of ActOn, focusing on its different software and knowledge components, and on the main interactions between them. Section 3 describes the prototype implementation for the EGEE Grid, which instantiates this architecture. Section 4 discusses related work, focusing on the main similarities and differences between ActOn and other ontology-based information integration approaches described in the literature. Finally, Section 5 provides conclusions, and describes open issues and our planned future work.

2 The Active Ontology Approach

2.1 Requirements for Active Ontology

The list of requirements presented in this section is based on the actual information integration needs that we have identified in dynamic, distributed systems like the EGEE Grid, Crossgrid, and Unicore [7–9].

- We need to deal with frequent changes of parts of the metadata, which are caused by the dynamic features of the entities of a large-scale distributed system.
- We need to have an efficient and economic way to avoid a continuous metadata update process, which is expensive for a large-scale distributed system.
- 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.
- We need to create/update the metadata that captures only those aspects that we are interested in.

Though these requirements have been obtained from our aim to develop an aggregated information service for the EGEE Grid infrastructure, similar requirements can be also found in other application domains such as the stock market or the currency exchange domain. For this reason, we have aimed at proposing a solution that is generic enough, so that in the future we will be able to apply it to other similar domains.

2.2 An Overview of the Active Ontology Architecture

ActOn is comprised of a set of software components, such as a metadata scheduler (MSch), an information source selector (ISS), a metadata cache (MC), and a set of information wrappers; and a set of knowledge components, which represent knowledge from the application domain and from the information sources.

Figure 1 shows how these components are interrelated and how they are related to the corresponding information sources that they take data from. The components are described briefly below (starting with the knowledge components), and in more detail in Section 2.3.

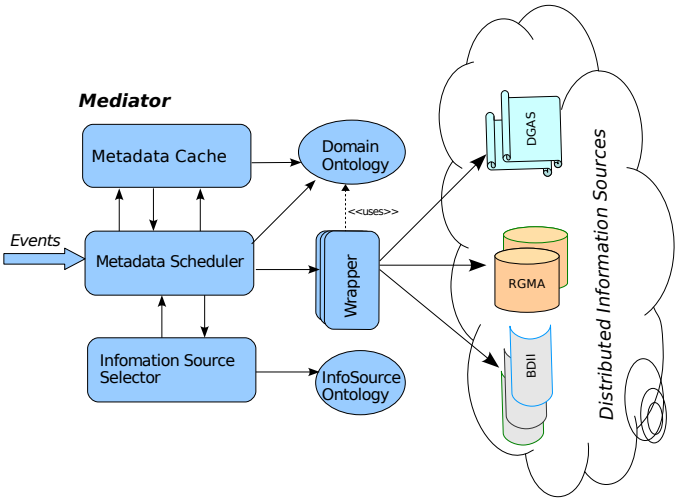

Fig. 1. The Overview of the Active Ontology Architecture

Knowledge Components

1. A (set of) domain ontology(ies). They describe the metadata information model in the form of domain concepts and properties for which instances will be generated. In our sample domain, presented in Section 3, they describe resources, components, services, and applications of the EGEE Grid.
2. An ontology of the information sources. It provides information about the characteristics of information sources, which are used for the information source selection process.

Both ontologies are related by means of mappings that specify which domain concepts and properties can be generated by which information sources.

Software Components

1. Metadata Scheduler. It is a policy-based metadata management service for managing and updating metadata. It decides if any metadata needs to be updated based on triggering events or on the metadata lifetime information.

2. Information Source Selector. It is used to find suitable information sources according to the real-time status of the system and to the information needs.
3. Information Wrappers. They are used to fetch information from heterogeneous information sources an transform it into the information model described by the set of domain ontologies.
4. Metadata Cache. It is used to maintain up-to-date metadata together with its lifetime information. It stores metadata into a repository and is in charge of managing queries over it.

2.3 The ActOn Components

Domain and Information Source Ontologies. The domain ontology(ies) define the global information model used to represent metadata, hence they are completely application dependant. We will give an example of one of these ontologies in Section 3. ActOn does not put any constraint about the language to be used to implement these ontologies, although in our current implementation we assume that ontologies are described either in RDF Schema or OWL.

The Information Source Ontology is designed to assist in locating suitable information sources for a specific information need. This ontology describes the features of the information sources to be used by the system, including their schema, information model, access APIs, access points, lifetime characteristics, etc. It is developed in OWL, and contains only five classes and forty properties. This domain-independent part has to be extended with domain-specific information source descriptions when it is used for a specific application. The most important class in this ontology is `InformationSource`, which is described with four properties:
(1) `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;
(2) `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";
(3) `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;
(4) `withSchema`: it indicates the kind of information that an information source provides. For instance, the EGEE BDII servers use the Glue Schema.

ActOn follows a global-as-view approach for information integration [10], that is, the global schema is expressed in terms of the data sources (every property for every class defined in the set of domain ontologies is associated to one or several information sources where the data can be obtained from). The association is expressed by linking the domain ontology components with the information source ontology classes. This is done by means of the property `generatedBy`, which represents not only this association but also the means to be used to extract information from the source and transform it into the domain ontology(ies) components. This is expressed with the class `Schema`. The transformation can be simple, such as a set of attribute-attribute pairs, or complex, such as an expression in a specific language to generate wrappers (as in approaches like WSL, D2R, R2O, etc [10–12]).

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 used, so as to

avoid unnecessary updates. We adopt event-driven mechanisms to cope with that policy. We have defined three types events that can trigger the update process, though we have only implemented the first one in our prototype. They are:

(i) 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, we do not contact the information sources. If not, then we contact them to get *fresh* metadata [2].
(ii) 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`.
(iii) Application-specific events. They 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.

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 it the metadata cache, assigns a time-stamp to it and sends back the results to the requestor.

Our approach has clear advantages over others that update metadata on a regular time-scale basis, such as the approaches followed by Globus MDS and gLite BDII. These systems keep updating all their metadata every 6-8 minutes. 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.

Information Source Selector (ISS). 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, there are over 20 region BDII servers which can be used to fetch the information about the EGEE Computing Elements.

Taking this into account, ActOn includes the *ISS*, which selects the most suitable information source among those available. The selection is based on a set of a retrieval conditions, including the actual information needed (specified as a SPARQL query), and other aspects like the geographical proximity of the source. An example of this selection process in the context of EGEE is provided in Section 3.

Information Wrappers. After an information source is selected, the Metadata Scheduler contacts the corresponding Information Wrappers in order to retrieve the relevant up-to-date information. Normally there is an Information Wrapper per type of information source

[2] 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

accessed (that is, one for MDS, another one for BDDII, etc.). 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 the selected 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. A more detailed example is shown in Section 3.

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, D2R, R2O, etc [10–12].

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.

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 Figure 2.

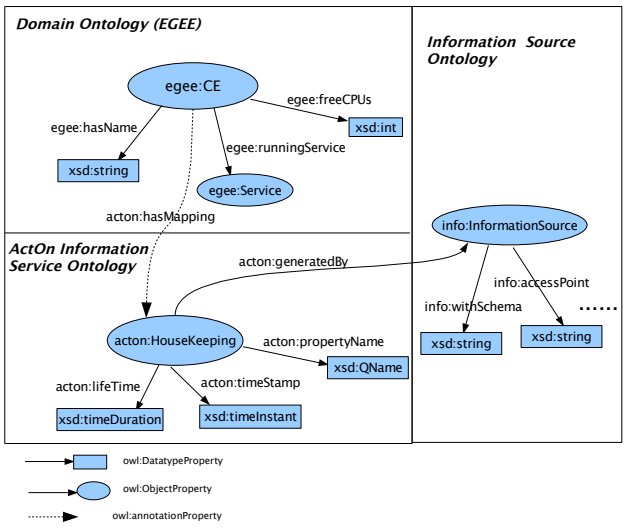

Fig. 2. Graphical overview of the association between domain and information source ontologies

3 A Case Study: Building an Information Aggregation Service for EGEE Grid

In this section, we illustrate how ActOn can be used to create an information aggregation system for EGEE. EGEE is a good example to show the benefits of using ActOn, because it poses the challenges that motivate our work: highly dynamic information sources that are geographically distributed.

We have implemented this service[3] to maintain metadata about EGEE computing and storage elements, and about virtual organisations. The service uses Globus Toolkit 4 (GT4) [14] and the OntoGrid Semantic Binding Service. The Semantic Binding Service is part of the S-OGSA middleware core services [15] and 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.

In the following subsections we provide details about the ontologies used in this prototype, which are the basis for configuring the software components described in the previous section.

3.1 The EGEE and gLite Middleware Domain Ontologies

The domain ontologies define the global information model used to represent metadata, hence they are completely application dependant. The Grid can be considered as a collection of Virtual Organizations (VOs) and of different kinds of resources. Resources are organized and utilized by Grid middleware to provide Grid users with computing power, storage capability, and services required for problem solving. VOs enable disparate groups of organizations and/or individuals to share resources in a controlled fashion, so that members may collaborate to achieve shared goals.

Therefore, we regard a Grid as a constellation of VOs, which includes VOs, users, applications, middleware, services, computing and storage resources, networks, and policies of use. As shown in Figure 3, the Grid domain model is layer-structured, and is designed around a simple three-layer scheme. The top layer includes the *Foundational Ontology*, *OGSA Onotlogy* and *S-OGSA Ontology*, and *Information Source Ontology*; Middleware Onotlogies lie on the middle layer; the bottom layer includes the service onotlogies.

We have created OWL ontologies (*gLite Ontology and EGEE Ontology*) that describe the EGEE infrastructure and its entities, including concepts like Computer Element, Storage Element, User Interface, Worker Node, Resource Broker, Logging and Booking Service, and Site. These ontologies are based on the one described in [16] and extend the Grid ontology described in [17], which include descriptions about virtual organisations, users, applications, middleware services, computing and storage resources, networks, and usage policies.

3.2 The EGEE Information Source Ontology

The basic information source ontology described in section 2.3 defines the concepts and properties used to describe any information source (this is the Information Source ontology in Figure 3). For our prototype, we have extended that ontology with the description 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,

[3] The source code can be downloaded from [13]

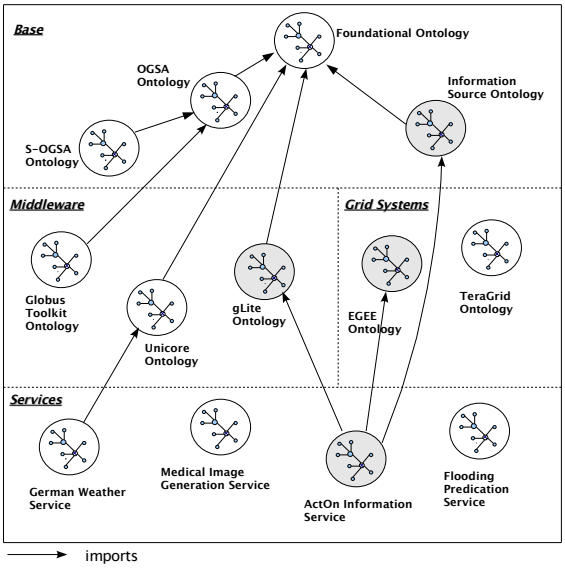

Fig. 3. An Overview of the ActOn Ontologies in the context of the Grid Ontology [17]

10 instances of RGMA, 5 GridICE, and 10 Unix-script. They can be found at [13], and are included in the ActOn Information Service Ontology in Figure 3.

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 Implementation of the ActOn Software Components for EGEE

The Metadata Scheduler (MSch) is implemented a service to receive different events, queue them, interacting with ISS. When a query event is triggered, for instance, which 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.

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'' . }
```

We have developed four kinds of wrappers for the four different information sources aforementioned: the BDII server wrapper, the RGMA server wrapper, the GridICE wrapper, and the Unix-script wrapper. These wrappers are invoked by the Metadata Scheduler (MSch), which provides the parameters that the wrappers need. For example, the BDII wrapper needs two parameters, accesspoint and attributes, in order to fetch the values requested. These values are obtained from the information source selection step. For instance, in the previous case the parameters sent will be an LDAP query like "ldapsearch -x -H ldap://prod-bdii.cern.ch:2170 -b mds-vo-name=CERN-PROD,o=grid".

We have developed the MC as an RDF triple cache for EGEE resources, such as Computing Element (CE), Storage Element (SE), Virtual Organisation (VO), using the Jena API [18]. The functions of the MC include: (i) placing/replacing RDF triples to MC; (ii) extracting required triples from the cache; (iii) removing expired cached triples. In particular, an RDF inference model [19] is built for the MC. RDF triples about heterogeneous Grid resources are added into the model by inserting a set of statements into the model.

4 Related Work

Many sets of criteria have been used for the classification of existing ontology-based information integration systems ([6, 20]). One sample criterion is the place where information resides, which allows us distinguishing between mediator and warehouse approaches [21], 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) [22–24] and the Global as View (GaV) [25] approaches. Others focus on the degree of automation of mappings between sources and ontologies [20].

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 [26]:

- 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).

Table 4 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 [27], and its successor Prometheus [28], are on-demand approaches focused on integrating data from many different types of information sources, 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 [29]). Prometheus also addresses identity reconciliation.
- Carnot [30], and its successor InfoSleuth [31], 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 [25] and Information Manifold [24] 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 [32], PICSEL [22] and TAMBIS [33] 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 [34] 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 [35] 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 hypothesis 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 Table 4 allow reasserting our initial assumption about the fact that none of the existing approaches are 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 those 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.

5 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

	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 (in progress)	No	Yes	No

Fig. 4. Features of the most common ontology-based information integration approaches

when dealing with highly dynamic, distributed and redundant information sources in the cases where response time is an important non-functional requirement.

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.

We have actually implemented a prototype of an information service for the EGEE Grid system, one of the largest production Grid systems in the world. This system has been described in section 3 and can be downloaded from [13].

As for the detailed evaluation, we have analysed the average and worst case time responses of our system with respect to other configurations where no metadata cache is used, as well as the accuracy of the information provided to the requestor with different alternatives, such as those based only on materialising information (with or without updates) and those based only on virtual information access on demand. We also designed experiments for information quality measurement and conducted them on the EGEE Grid testbed. The experiment results show : 1) BDII has bad precision results for complex queries because of its weak query (LDAP-based query) ability; 2)RGMA is very sensitive to the registering and availability of information providers at a given point in time; 3) Some complex queries cannot be answered by BDII or RGMA in isolation; 4) 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. The details are presented in [36].

For the future work, we plan to work on several aspects of our system, including the integration of features and technologies from other similar systems, and the application to other scenarios with similar requirements. One on-going work is the deployment of the ActOn-based information service on a Grid testbed so that people can use it to query for the EGEE Grid resources in reality.

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 [29], with the metadata cache approach that we have proposed.

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 by the Marie Curie fellowship RSSGRID (FP6-2002-Mobility-5-006668), and by the EU FP6 CoreGrid Network of Excellence (FP6-004265). We also thank Pinar Alper, Antun Balaz and Laurence Field (EGEE porject), Georges Da Costa and Anastasios Gounaris (CoreGrid WP2), for their helpful comments.

References

1. "EGEE gLite," http://glite.web.cern.ch/glite.
2. I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, and J. V. Reich, *The Open Grid Services Architecture, Version 1.5*, gfd-i.080 ed., GGF, July 2006, http://forge.gridforum.org/projects/ogsa-wg.
3. "Berkeley Database Information Index (BDII)," http://lfield.home.cern.ch/lfield/cgi-bin/wiki.cgi?area=bdiipage=documentation.
4. K. Czajkowski, S. Fitzgerald, I. Foster, and 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, August 2001.
5. "EDG RGMA," www.marianne.in2p3.fr/datagrid/documentation/rgma-guide.pdf.
6. H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner, "Ontology-based integration of information — a survey of existing approaches," in *IJCAI–01 Workshop: Ontologies and Information Sharing*, H. Stuckenschmidt, Ed., 2001, pp. 108–117.
7. "Enabling Grids for E-sciencE (EGEE)," http://public.eu-egee.org/.
8. J. Marco and et al., "First Prototype of the Crossgrid Testbed," in *Proceedings of First European AcrossGrids Conference (AXGrids 2003), LNCS 2970*. Santiago de Compostela, Spain: Springer-Verlag, 2003, pp. 67–77.
9. P. Wieder and D. Mallmann, "UniGrids - Uniform Interface to Grid Services," in *7th HLRS Metacomputing and Grid Workshop*, Stuttgart, Germany, April 2004.
10. H. Garcia-Molina, Y. Papakonstantinou, A. R. a. D. Quass, Y.Sagiv, J. Ullman, V. Vassalos, and J. Widom, "The TSIMMIS Approach to Mediation: Data Models and Languages," *Intelligent Information Systems*, vol. 8, no. 2, pp. 117–132, 1997.
11. J. Barrasa, O. Corcho, and A. Gomez-Perez, "R2O, an Extensible and Semantically based Database-to-Ontology Mapping Language," in *In Proceedings of the 2nd Workshop on Semantic Web and Databases(SWDB2004)*, Toronto, Canada, 2004.
12. C. Bizer, "D2R MAP: A DB to RDF Mapping Language," in *12th International World Wide Web Conference*, Budapest, May 2003.
13. "OntoGrid CVS," http://www.ontogrid.net/ontogrid/downloads.jsp.
14. "Globus toolkit," http://www.globus.org.
15. Ó. Corcho, P. Alper, I. Kotsiopoulos, P. Missier, S. Bechhofer, and C. A. Goble, "An Overview of S-OGSA: A Reference Semantic Grid Architecture," *Journal of Web Semantics*, vol. 4, no. 2, pp. 102–115, 2006.
16. W. Xing, M. D. Dikaiakos, and 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)*. Singapore: IEEE Computer Society, May 2006, pp. 178–184.
17. M. Parkin, S. van den Burghe, O. Corcho, D. Snelling, and J. Brooke, "The Knowledge of the Grid: A Grid Ontology," in *Proceedings of the 6th Cracow Grid Workshop*, Cracow, Poland, October 2006.
18. "Jena: A Semantic Web Framework for Java," http://jena.sourceforge.net/.
19. G. Moore and A. Seaborne, "RDF Net API," W3C Member Submission, Tech. Rep., 2003.
20. L. Bellatreche, D. N. Xuan, G. Pierra, and H. Dehainsala, "Contribution of Ontology-based Data Modeling to Automatic Integration of Electronic Catalogues within Engineering Databases," *Computers in Industry Journal*, pp. 711–724, 2006.
21. J. D. Ullman, "Information integration using logical views," in *ICDT '97: Proceedings of the 6th International Conference on Database Theory*. London, UK: Springer-Verlag, 1997, pp. 19–40.
22. F. G. V. Lattes and M.-C. Rousset, "The Use of CARIN Language and Algorithms for Information Integration: The PICSEL System," *International Journal of Cooperative Information Systems*, vol. 9, no. 4, pp. 383–401, 2000.
23. C. Reynaud and G. Giraldo, " An application of the mediator approach to services over the Web ," in *Concurrent Engineering*, J. Cha, R. Jardin-Gonçalvez, and A. Steiger-Garçao, Eds., vol. 1. A.A. Balkema Publishers, July 2003, pp. 209–216.

24. A. Y. Levy, A. Rajaraman, and 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.
25. S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. D. Ullman, and 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.
26. M. Michalowski, J.-L. Ambite-Molina, S. Thakkar, R. Tuchinda, C. Knoblock, and S. Minton, "Retrieving and semantically integrating heterogeneous data from the web," *IEEE Intelligent Systems*, vol. 19, no. 3, pp. 72–79, 2004.
27. Y. Arens, C. A. Knoblock, and W.-M. Shen, "Query reformulation for dynamic information integration," *J. Intell. Inf. Syst.*, vol. 6, no. 2-3, pp. 99–130, 1996.
28. L. Padgham and 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.
29. G. Barish, D. DiPasquo, C. A. Knoblock, and S. Minton, "Dataflow plan execution for software agents," in *Proceedings of the Fourth International Conference on Autonomous Agents*, C. Sierra, M. Gini, and J. S. Rosenschein, Eds. Barcelona, Spain: ACM Press, 2000, pp. 138–139.
30. M. P. Singh, P. Cannata, M. N. Huhns, N. Jacobs, T. Ksiezyk, K. Ong, A. P. Sheth, C. Tomlinson, and D. Woelk, "The carnot heterogeneous database project: Implemented applications," *Distributed and Parallel Databases*, vol. 5, no. 2, pp. 207–225, 1997.
31. 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, and 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. New York: ACM Press, 13–15 1997, pp. 195–206.
32. E. Mena, A. Illarramendi, V. Kashyap, and 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)*, vol. 8, no. 2, pp. 223–272, April 2000, kluwer Academic Publishers.
33. C. Goble, R. Stevens, G. Ng, S. Bechhofer, N. Paton, P. Baker, M. Peim, and A. Brass, "Transparent Access to Multiple Bioinformatics Information Sources," *IBM Systems Journal*, vol. 40, no. 2, pp. 534–551, 2001.
34. M. A. Jeusfeld, C. Quix, and 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.
35. 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, and I. Martos, "A semantic portal for the international affairs sector." in *EKAW*, 2004, pp. 203–215.
36. W. Xing, O. Corcho, C. Goble, and M. Dikaiakos, "Information Quality Evaluation for Grid Information Services," submitted to CoreGrid Symposium in conjunction with Euro-Par 2007.