A Decentralized Infrastructure for Query Answering over Distributed Ontologies

Peter Haase and Yimin Wang
Institute AIFB, University of Karlsruhe (TH)
Karlsruhe, 76128, Germany
{pha,ywa}@aifb.uni-karlsruhe.de

ABSTRACT

In this paper we describe an infrastructure for query answering over distributed ontologies on the Semantic Web. This infrastructure addresses (i) the coordination of multiple nodes using metadata about the provided resources managed in a decentralized registry and (ii) the mediation between heterogeneous ontologies via an expressive mapping formalism along with corresponding reasoning algorithms for query answering. Our approach is based on a virtual integration that exhibits a semantics as if all ontologies were integrated locally. Practically, the distributed ontologies still reside on the remote peers, and only the parts relevant for answering the query need to be retrieved to the local node. Experimental evaluations with the implementation in KAONp2p show that the approach is very promising, as the performance of query answering is essentially dominated by the size of the data, and only slightly affected by the degree of distribution and heterogeneity.

1. INTRODUCTION

The realization of real-life applications in the Semantic Web requires the ability to deal with heterogeneous ontologies fragmented and distributed over multiple autonomous nodes. In recent years much progress has been made in providing efficient and scalable reasoning support over expressive ontologies: We now have a number of reasoners such as RacerPro[1], Pellet[2] [20], KAON2[3] that allow to handle ontologies with reasonable size and complexity. However, these reasoners typically assume centralized and closed settings, where a number of known ontologies are integrated in one local node. We argue that a decentralized infrastructure better reflects the spirit of the open Semantic Web, where ontologies are distributed over a number of autonomous nodes. When dealing with such decentralized infrastructures, we need to consider a number of arising challenges that are currently not addressed in centralized reasoners. The first fundamental challenge is the *coordination of autonomous nodes*, i.e. the ability to manage the organization of the interaction between nodes. In the case of completely centralized architectures, one node has complete control over all other nodes, whereas in completely decentralized architectures there is no central control and all nodes act autonomously. A particularly important coordination task is that of discovering and selecting resources relevant for answering queries as well as routing of requests: How do you find the right nodes that are able to answer a given query in a decentralized system in a scalable manner without any centralized servers or hierarchy? Related to the problem of autonomy is that of *heterogeneity*: To enable interoperability between nodes in large distributed information systems based on heterogeneous ontologies, it is necessary to specify how the ontologies residing at a particular node correspond to ontologies residing at another nodes. It is also necessary to formally define the notion of a mapping between ontologies. Finally, we need *reasoning* algorithms that can efficiently deal with ontologies distributed over multiple nodes, taking into account the semantics of the mappings between the individual ontologies. For efficiency reasons it is important to devise methods that do not require the integration of ontologies in a single node, but that only retrieve the information relevant for the particular reasoning task.

In this paper we propose an infrastructure, in which we address these challenges in an integrated manner to realize query answering over heterogeneous OWL DL ontologies distributed over multiple nodes. This infrastructure builds on, extends and integrates a number of prior techniques we have developed in the context of distributed metadata management and reasoning. In order to deal with the coordination of the nodes we rely on metadata about nodes and their provided resources to support their interoperation and coordination. In our approach, nodes advertise descriptions of their resources and can thus establish acquaintances with other nodes. Each node maintains the metadata about available resources in its own local registry in a completely decentralized manner. Acquainted nodes can then share data and coordinate their interaction. To be able to integrate heterogeneous ontologies, we propose an expressive mapping formalism in which mappings are expressed as correspondences between queries over source and target ontologies. An important aspect of this mapping formalism is that it does not rely on the notion of a global ontology, as known in classical integration systems based on LAV or GAV (Local-As-View, Global-As-View). Instead, mappings can be formulated in any direction between arbitrary nodes. We have implemented the reasoning infrastructure in KAONp2p, extending the OWL reasoner KAON2. The paper is organized as follows: In Section 2 we discuss several dimensions of related work. In Section 3 we present a general overview of the KAONp2p infrastructure. In the subsequent sections we discuss the aspects of metadata and resource selection (Section 4), the mapping formalism and the algorithms for query

[1] http://www.racer-systems.com/
[2] http://www.mindswap.org/2003/pellet/
[3] http://kaon2.semanticweb.org/

SAC'07 March 11-15, 2007, Seoul, Korea

answering (Section 5). In Section 6 we present evaluation results. We conclude with an outlook to future work in Section 7.

2. RELATED WORK

There are several threads of research work that are related to our work: (1) The use of semantics and metadata and corresponding registries for the coordination and organization of distributed information systems, (2) representation of mappings for ontology integration, and (3) approaches to distributed reasoning.

In the past, there have been various proposals for modeling metadata of ontologies. Unfortunately, none of them has been accepted as a standard, some proposals, such as Dublin Core, were too general, others were limited in applicability. [14] has proposed an ontology meta-ontology (OMO) for a distributed ontology registry. The focus of the registry is on locating, re-using and evolving existing ontologies rather than supporting particular reasoning tasks. Semantic representation of resources have further been successfully applied to organize distributed systems with *semantic overlay networks*. In these semantic overlay networks, links are created according to semantic relationships between the nodes. The neighborhood thus mirrors semantic relationships between the peers. For example, Gridvine [1] uses the semantic overlay for managing and mapping data and metadata schemas, on top of a physical layer consisting of a structured Peer-to-Peer overlay network called P-Grid. A similar approach is taken in our infrastructure, where nodes maintain a registry with metadata about acquainted nodes, which is used to select relevant nodes and ontologies for answering a given query.

The task of *ontology integration* using mappings is very related to that of data integration in databases. In [13] the author introduces a general framework for data integration and compares existing approaches to data integration (GAV, LAV) along this framework. The work on data integration has been extended and re-applied to ontology integration in [3]. Here the authors follow the classical distinction between LAV and GAV approaches and outline query answering algorithms for these specific settings. In contrast to this work, query answering in our ontology integration system is not bound to these restricted forms of mappings.

With respect to query processing over distributed data, there is a large amount of related work in the area of Peer-to-Peer databases, such as [21], [2], [5]. However, most of the work in Peer-to-Peer databases assumes that queries can be answered by simply forwarding the query to other nodes and aggregate the answers afterwards. Such an approach is not sufficient for distributed ontologies. Consider the simple example of two knowledge bases $K_1 = \{Student \sqsubseteq Person\}$ and $K_2 = \{Student(paul)\}$. Evaluating the query $Q(x) := Person(x)$ against either K_1 or K_2 will return no result; only the combination of the knowledge of K_1 and K_2 will return the desired result. In description logics terminology, Peer-to-Peer databases only allow to deal with distributed A-Boxes. On the other hand, reasoners for distributed description logics such as Drago [19] currently provides no support for handling assertional knowledge at all.

3. OVERVIEW OF KAONP2P

In this section, we present a brief overview of KAONp2p[4]. The general architecture of a single KAONp2p node interacting with remote nodes is shown in Figure 1. In the following, we discuss the individual components of the system architecture.

[4]The system is freely available for download at `http://kaonp2p.ontoware.org/`.

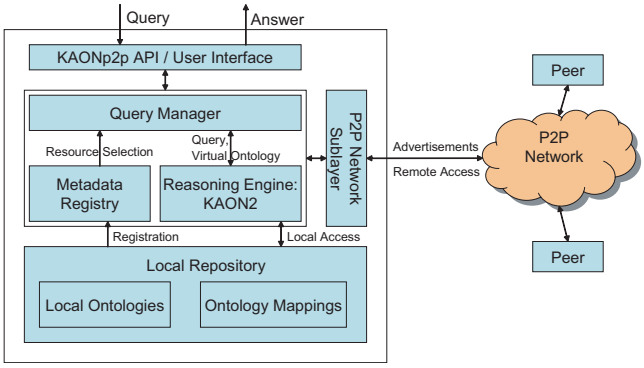


Figure 1: Overview of KAONp2p Architecture

The *Local Repository* of a node contains the ontologies it provides to the network along with mappings that relate heterogeneous ontologies available in the network. It is important to note that mappings are first-class objects in the system that can be shared with other nodes.

The *Query Manager* is the component responsible for answering queries against the available ontologies in the network. As queries we consider conjunctive queries over OWL DL ontologies. The query process can be divided into two steps:

1. **Resource selection.** The goal of the resource selection is to identify resources in the network that are relevant to answer a particular user query. This process is governed by selection algorithm that matches the subject of the query against resource descriptions stored in the *Metadata Registry*. The Metadata Registry maintains metadata about resources available (i.e. peers, ontologies, and mappings), which may be accessible either locally or remotely in the network. The resources are described using the metadata ontology described in Section 4. The result of the selection process is a "virtual ontology" that logically integrates relevant ontologies and mappings required to mediate between the heterogeneous ontologies in the network, represented using the mapping formalism described in Section 5.
2. **Query answering.** In the second step, the query is evaluated against the virtual ontology within the *Reasoning Engine*. In our implementation, we rely on KAON2 as a reasoner. The reasoning algorithms of KAON2 do not require the integrated ontologies to be materialized locally, instead the distributed ontologies still reside on the remote server, and only relevant parts need to be retrieved to the local node. The details of this process will be explained in Section 5.

The *Peer-to-Peer network sub-layer* provides communication services for the data exchange with remote nodes, i.e. to propagate advertisement messages and to realize the access to remote ontologies. In the implementation of KAONp2p, we rely on an RMI-based implementation, however, other communication protocols would be possible as well.

By means of the *KAONp2p user interface and API*, users can pose queries, receive answers to queries and control the configuration of the system. While in the user interface queries are formulated in a visual manner, they are passed to the API as conjunctive queries in SPARQL.

4. METADATA FOR DISTRIBUTED ONTOLOGIES

In this section we present a brief overview of our ontology for the representation of metadata about nodes in the network and the

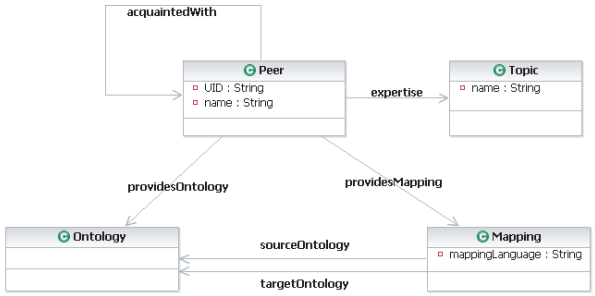


Figure 2: Overview of the P-OMV Ontology

resources they provide, i.e. the ontologies. We also show how the task of resource selection is realized using this metadata ontology. In this approach, the nodes advertise descriptions of their resources and can thus establish acquaintances with other nodes. Acquainted nodes can then share data and coordinate their interaction.

4.1 Ontology and Peer Metadata

For the description of ontology metadata we rely on OMV, the Ontology Metadata Vocabulary [10]. In the following we provide an overview of the main properties of the OMV ontology with a focus on the properties relevant for the problem of resource discovery and selection. For a complete reference and the complete ontology we refer the reader to `http://omv.ontoware.org/`. We model various types of metadata of ontologies, which we can classify as *Descriptive metadata*, *Provenance metadata* about the creation process, *Dependency metadata* managing relationships with other ontologies such as compatibility, and *Statistical metadata*, e.g. about the size of the ontology in terms of ontology elements, axioms etc.

Descriptive metadata is the most important type of metadata for the discovery and selection of ontologies. Descriptive metadata relates to the domain modeled in the ontology in form of keywords, topic classifications, textual descriptions of the ontology contents etc. It includes the `name` by which the ontology is known, the `language`, its `type` (e.g. *top-level*, *core*, *task*, *domain*, and *application ontology*), and the `subject`: The subject of an ontology provides a classification in terms of the domain. The subject is expressed as a classification against established topic hierarchies, such as the general purpose topic hierarchy DMOZ[5] or the domain specific ACM topic hierarchy[6] for the computer science domain.

Besides the ontology themselves, the second important resources to describe are the nodes managing and providing these informational resources, which we call *peers* in our metadata ontology. The extensions required to model metadata of peers are realized as an extension to the OMV ontology, called P-OMV. Figure 2 shows an overview of the P-OMV ontology. The metadata required to describe peers include descriptive information about the peers themselves, their relationship with other peers, as well as information about the resources they provide:

Each peer carries a unique ID (`UID`) to be identified. Depending on the underlying communication infrastructure of the network sublayer, different addressing schemes may be applied. In our implementation of KAONp2p, we simply use IP addresses. In addition to the unique identifier, each peer carries a `name` for identification, which is primarily used for human interpretation. The `expertise` is an abstract description of the peer in terms of some topic ontology. Depending on the application scenario, the expertise of the peer can be the subjects of the ontologies that the peer provides, or a more generic description of expertise.

The property `acquaintedWith` describes the acquaintances of a peer with other peers. The Peer-to-Peer network then consists of local peers, each with a set of acquaintances, which define the Peer-to-Peer network topology. The property `providesOntology` describes the relationship between the peer and the ontologies provided by the peer. It is essential for locating relevant information resources in the network.

The property `providesMapping`: is used to describe which mappings between ontologies a peer provides. Mappings are used to describe the correspondences between different ontologies provided by the peers. The properties `sourceOntology` and `targetOntology` specify the ontologies that are being mapped. In general, mappings need not be symmetric, a distinction between mapping source and target is therefore required. The property `mappingLanguage` is used to indicate the language that is used to express the mapping. In KAONp2p we rely on the formalism for ontology mappings presented in Section 5, which can be expressed in SWRL. However, other languages may be used for the representation of ontology mappings.

4.2 Discovery and Selection of Resources

In our approach to resource discovery and selection we follow the successful approach of expertise-based peer selection [9], which have already been applied in the Peer-to-Peer systems Bibster [7] and Oyster [17]. In this approach, peers advertise their resource descriptions according to the metadata ontology in the network to form acquaintances. Communication autonomy implies that peers are fully autonomous in choosing their acquaintances. Moreover, we assume there is no global control in the form of a global registry to manage acquaintances. Acquaintances are managed in a decentralized manner, i.e. by the individual peer using its metadata registry. For the selection of resources we offer three options: (1) a manual selection, where the user can choose the resources relevant to his query, (2) a trivial selection that includes all resources known in the registry, and (3) an automated selection of resources based on matching the subject of a query and the expertise according to their semantic similarity, which serves as an indicator for relevance. A subject is an abstraction of a given query expressed in a set of terms from the metadata ontology. The subject can be seen as a complement to an expertise description, as it specifies the required expertise to answer the query. We rely on the notion of a similarity function as defined in [4]. The similarity function determines the semantic similarity between a subject and an expertise description. As such, an increasing value indicates increasing similarity and relevance. The resource selection algorithm returns a ranked set of resources, where the rank value is equal to the similarity value provided by the similarity function. For example, to answer a query about the subject of *Databases*, the resource selection might identify a peer who provides an ontologies about the subject of the ACM topic is *Information Systems / Database Management* as relevant. For the details of the selection process, we refer the reader to [9]. In addition to the selected ontologies, available mappings are identified that relate the heterogeneous remote ontologies to the target ontology against which the query is expressed. The relevant resources are integrated in the virtual ontology, which will be used in the second step of query answering described in the following sections.

[5] `http://dmoz.org/`
[6] `http://www.acm.org/class/`

5. MAPPINGS AND QUERY ANSWERING OVER DISTRIBUTED ONTOLOGIES

To enable interoperability between nodes in large distributed information systems based on heterogeneous ontologies, it is necessary to specify how the data residing at a particular node corresponds to data residing at another node. This is formally done using the notion of a mapping. There are three lines of work connected to the problem of mapping: (1) identifying correspondences between heterogeneous data sources, (2) representing these correspondences in an appropriate mapping formalism, and (3) using the mappings for a given integration task. We here assume that the correspondences between data sources are already known and focus on the latter two important problems: those of representing the mappings using an appropriate formalism and using them for the task of query answering over heterogeneous data sources.

We follow the general framework of [13] to formalize the notion of a mapping system for OWL DL ontologies, where mappings are expressed as correspondences between conjunctive queries[7] over ontologies. The components of this mapping system are the source ontology, the target ontology, and the mapping between them.

DEFINITION 1 (OWL DL MAPPING SYSTEM). *An OWL DL mapping system $\mathcal{MS}$ is a triple $(\mathcal{S}, \mathcal{T}, \mathcal{M})$, where*

- *$\mathcal{S}$ is the* source OWL DL ontology*, $\mathcal{T}$ is the* target OWL DL ontology*,*
- *$\mathcal{M}$ is the mapping between $\mathcal{S}$ and $\mathcal{T}$, i.e. a set of assertions $q_S \rightsquigarrow q_T$, where q_S and q_T are conjunctive queries over $\mathcal{S}$ and $\mathcal{T}$, respectively, with the same set of distinguished variables $\mathbf{x}$, and $\rightsquigarrow \in \{\sqsubseteq, \sqsupseteq, \equiv\}$.*

An assertion $q_S \sqsubseteq q_T$ is called a sound *mapping, requiring that q_S is contained by q_T w.r.t. $S \cup \mathcal{T}$; an assertion $q_S \sqsupseteq q_T$ is called a* complete *mapping, requiring that q_T is contained by q_S w.r.t. $\mathcal{S} \cup \mathcal{T}$; and an assertion $q_S \equiv q_T$ is called an* exact *mapping, requiring it to be sound and complete.*

Let us discuss the expressiveness in terms of the ontology language, the query language and the assertions. The expressiveness of conjunctive queries corresponds to that of the well-known select-project-join queries in relational databases. Two typical approaches to specify mappings are the *global-as-view* (GAV) approach, where elements of the target are described in terms of queries over source, and the *local-as-view* (LAV) approach, where elements of the source are described in terms of queries over target. Our mapping system subsumes the approaches of GAV, LAV. In fact, it corresponds to the GLAV approach, which is more expressive than GAV and LAV combined.

In [8] the semantics of the mapping system has been defined by translation into first-order logic. We here only discuss the intuitions behind the semantics of the main inference task for $\mathcal{MS}$, i.e. computing answers for a conjunctive query $Q(\mathbf{x}, \mathbf{y})$ w.r.t. $\mathcal{MS}$. To understand the intuition of computing answers, we briefly recall the semantics of query answering as defined in [13]: An answer of a conjunctive query $Q(\mathbf{x}, \mathbf{y})$ w.r.t. a knowledge base KB is an assignment θ of individuals to distinguished variables, such that $KB \models Q(\mathbf{x}\theta, \mathbf{y})$. Thus, the intuitive reading of this semantics is that an answer of a query needs to be entailed by the source ontology $\mathcal{S}$, the target ontology $\mathcal{T}$ and the mappings $\mathcal{M}$. This semantics is equivalent to the usual model theoretic semantics (e.g. in [3]) based on local and global models, where a query answer must be an answer in *every* global model.

As query answering within such a mapping system is undecidable in this generality, we have identified classes of mappings that introduce restrictions required to attain decidability. These restricted, but still very expressive mappings, can be expressed either directly in OWL DL, or in OWL DL extended with the so-called *DL-safe* subset of the Semantic Web Rule Language (SWRL) [16].

The first class of mappings captures the mappings that can be directly expressed in OWL DL. This is the case if q_s and q_t are of the form $P_s(\mathbf{x})$ and $P_t(\mathbf{x})$, where P_s and P_t are DL predicates: If q_s and q_t are of the form $P_s(x)$ and $P_t(x)$ and P_s, P_t are DL concepts, the mapping corresponds to the equivalent concept inclusion axiom. If q_s and q_t are of the form $P_s(x_1, x_2)$ and $P_t(x_1, x_2)$, with P_s and P_t are abstract or concrete roles, the mapping corresponds to the equivalent role inclusion axiom.

The second class of mappings captures the so-called *DL-safe Mappings*. Let us consider a sound mapping $q_S \sqsubseteq q_T$[8] with the assertion $\forall \mathbf{x} : q_T(\mathbf{x}, \mathbf{y}_T) \leftarrow q_S(\mathbf{x}, \mathbf{y}_S)$. In our restriction, we disallow the use of non-distinguished variables in the query q_T, i.e. restrict the assertions to the form $\forall \mathbf{x} : q_T(\mathbf{x}) \leftarrow q_S(\mathbf{x}, \mathbf{y}_S)$ [9] and require the query q_S to be DL-safe, thus limiting the applicability of the rules to known individuals. Thus obtained mappings correspond to (one or more) DL-safe rules, for which efficient algorithms for query answering are known [16].

We now show how to use an OWL DL mapping system for query answering in an *ontology integration system*, whose main task is to provide integrated access to a set of distributed source ontologies. The integration is realized via a mediated target ontology through which we can query the local ontologies.

DEFINITION 2. *For a set of local source ontologies $\mathcal{S}_1, \ldots, \mathcal{S}_n$, a target ontology $\mathcal{T}$ and corresponding mapping systems $\mathcal{MS}_1, \ldots, \mathcal{MS}_n$ with $\mathcal{MS}_i = (\mathcal{S}_i, \mathcal{T}, \mathcal{M}_i)$, an* ontology integration system *$\mathcal{IS}$ is again a mapping system $(\mathcal{S}, \mathcal{T}, \mathcal{M})$ with $\mathcal{S} = \bigcup_{i \in \{1 \ldots n\}} \mathcal{S}_i$ and $\mathcal{M} = \bigcup_{i \in \{1 \ldots n\}} \mathcal{M}_i$. The main inference task for $\mathcal{IS}$ is to compute answers of $Q(\mathbf{x}, \mathbf{y})$ w.r.t. $\mathcal{S} \cup \mathcal{T} \cup \mathcal{M}$, for $Q(\mathbf{x}, \mathbf{y})$ a conjunctive query over $\mathcal{T}$.*

Please note that because of the absence of a global ontology, this form of ontology integration system can be directly applied to decentralized integration: For our set of autonomous nodes, each relying on some local ontology, and a set of mappings that relate the local ontology to those of other nodes, an ontology integration system $\mathcal{IS} = (\mathcal{S}, \mathcal{T}, \mathcal{M})$ can easily be constructed for each individual node, where $\mathcal{S}$ consists of the ontologies of the remote node to be integrated, $\mathcal{T}$ is the ontology of the local node, and $\mathcal{M}$ consists of the individual mappings systems describing the correspondences between the local ontology with remote ontologies. This construction is performed during the resource selection process described in Section 4.2, which selects the relevant source ontologies $\mathcal{S}$ and the required mappings $\mathcal{M}$.

We now discuss how to compute answers to a conjunctive query $Q(\mathbf{x}, \mathbf{y})$ in an ontology integration system $\mathcal{IS}$. The algorithm is based on the correspondence between description logics and disjunctive datalog from [11], which is implemented in the KAON2 reasoner. Given an OWL DL knowledge base KB (without nominals) extended with DL-safe rules, a positive disjunctive datalog program $\mathsf{DD}(KB)$ is produced, which entails exactly the same set of ground facts as KB, i.e. $KB \models A$ if and only if $\mathsf{DD}(KB) \models A$, for A a ground fact. Thus, query answering in KB is reduced to query answering in $\mathsf{DD}(KB)$, which can be performed efficiently using the techniques of (disjunctive) deductive databases. Query answering can be performed in time exponential in the size of KB. Furthermore, as shown in [12], the data complexity of these algo-

[7] We denote a conjunctive query as $q(\mathbf{x}, \mathbf{y})$, with $\mathbf{x}$ and $\mathbf{y}$ sets of *distinguished* and *non-distinguished* variables, respectively.

[8] For a complete mapping $q_S \sqsupseteq q_T$, the situation is analogous, with the roles of q_S and q_T reversed.

[9] Please note that these assertions correspond to SWRL rules.

rithms (i.e. the complexity assuming the size of the schema is fixed) is NP-complete, or even P-complete if disjunctions are not used.

Based on these results, we are able to perform query answering in the ontology integration system by converting it into a disjunctive datalog program. The source ontology, target ontology and the mappings are converted into a disjunctive datalog program, and the original query is answered in the obtained program $\mathsf{DD}(\mathcal{S} \cup \mathcal{T} \cup \mathcal{M})$. By the results from [11, 16], it is easy to see that the algorithm exactly computes the answer of $Q(\mathbf{x}, \mathbf{y})$ in $\mathcal{IS}$.

From the above definition, one might get the impression that our algorithm requires that all source and target ontologies must be physically integrated into one mapping system in order to answer queries. This is, of course, not the case. More concretely, to compute $\mathsf{DD}(\mathcal{S} \cup \mathcal{T} \cup \mathcal{M})$, it is necessary to physically integrate the TBox part of $\mathcal{S}$, $\mathcal{T}$ and $\mathcal{M}$. Since the TBox are typically much smaller than the data, this does not pose practical problems. Accessing actual data sources (i.e. the ABoxes) is then governed by the chosen strategy for evaluating the datalog program. In practice, we only need to access the extensions of those predicates from remote nodes that are actually relevant for the datalog program.

6. EVALUATION

In this section we present experimental results for the evaluation of the KAONp2p infrastructure. In this evaluation we focus on the second part of the query processing, i.e. the actual query answering against a set of relevant resources. Please note that for this second part, the number of nodes will be typically much smaller than the number of nodes under consideration for the resource selection. For evaluation regarding the first step of resource selection (with several thousands of nodes), we refer to evaluations reported in [7] and [9].

In our first experiment we have used the Lehigh University Benchmark (LUBM) [6] to evaluate the performance and scalability in terms of the distribution. We deployed eight physically distributed KAONp2p nodes, each of which holding a different automatically generated data set according to the LUBM ontology[10] (describing instance data of one university, approximately 8MB of OWL/RDF data). We selected three typical SPARQL queries from the LUBM benchmark of different complexity:

```
* SELECT ?x WHERE { ?x rdf:type ub:GraduateStudent }
* SELECT ?x ?y WHERE { ?x rdf:type ub:AssistantProfessor .
  ?y rdf:type ub:Publication . ?y ub:publicationAuthor ?x}
* SELECT ?x ?y ?z WHERE { ?x rdf:type ub:GraduateStudent .
  ?y rdf:type ub:University . ?z rdf:type ub:Department .
  ?x ub:memberOf ?z . ?z ub:subOrganizationOf ?y .
  ?x ub:undergraduateDegreeFrom ?y }
```

We then used an additional node to perform the queries against the knowledge bases of 1..8 manually selected nodes. Figure 3 shows the results of the execution times for the query answering. The main observation is that in this particular scenario, the time for answering queries increases approximately linearly with the number of nodes and thus the size of the data set. The additional degree of distribution does not incur a performance penalty.

In a second experiment we evaluated the additional costs introduced by the heterogeneity between the nodes. In this experiment, we deployed two nodes, one with an automatically generated LUBM data set, another one with an SWRC[11] data set, containing real life data from the University of Karlsruhe[12]. Further, we defined mappings according to our mapping formalism to relate the LUBM ontology with the SWRC ontology in both directions. We then used an additional node to perform queries against the node providing the SWRC data set as source ontology, where in a first case the query is expressed in terms of the same target ontology in a homogeneous setting (SWRC - SWRC)[13] and in a second case the query is expressed against a different target ontology (LUBM - SWRC) in a heterogeneous setting. We repeated the experiment queries against the node providing the LUBM data set as source ontology, which we again queried with LUBM as target ontology (LUBM - LUBM) and SWRC as target ontology (SWRC - LUBM). Figure 4 shows the results for the query execution times. We observe that the time needed for query answering increases only slightly for the case where the source and target ontologies differ and thus mappings are required. The reason lies in the fact that the mappings are only used in the computation of the datalog program, which is neglectable compared to the evaluation of the program. This makes our approach especially applicable for scenarios where mappings between heterogeneous ontologies are required.

Summarizing, the evaluation results show that in our approach the performance of query answering is essentially dominated by the size of the data, and only slightly affected by the degree of distribution and heterogeneity. In fact, it shows a performance comparable to a setting where data resides on a single, homogeneous node.

Figure 3: Evaluation Results for LUBM 1..8

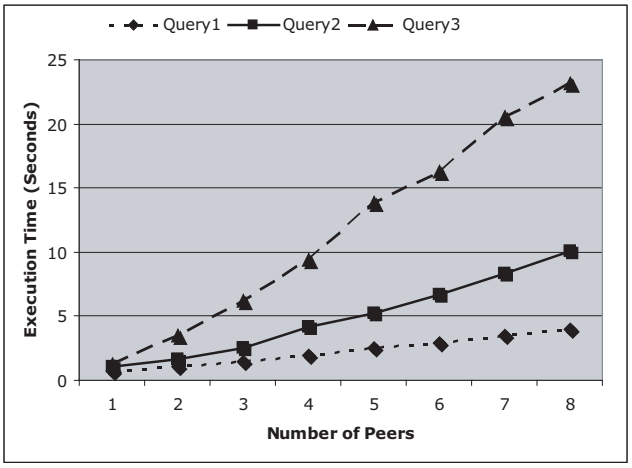
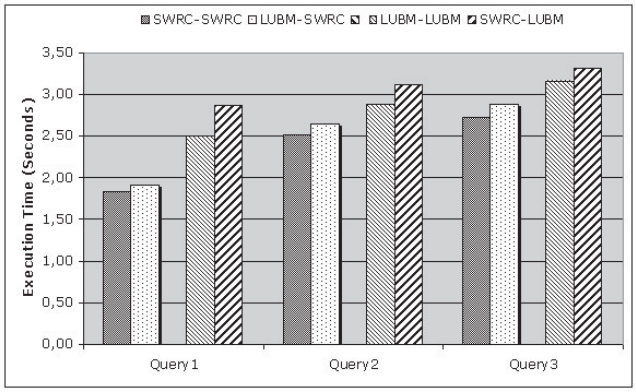


Figure 4: Cost of Mappings between Heterogeneous Ontologies



7. CONCLUSIONS AND FUTURE WORK

In this paper we have introduced KAONp2p, a Peer-to-Peer system for query answering over distributed ontologies in decentralized networks. This infrastructure addresses (i) the coordination of multiple nodes using metadata about the provided resources managed in a decentralized registry, (ii) the mediation between hetero-

[10] http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl

[11] http://ontoware.org/projects/swrc/

[12] http://www.aifb.uni-karlsruhe.de/viewAIFB_OWL.owl

[13] For SWRC as target ontology we rephrased the three queries above in terms of SWRC.

geneous ontologies via an expressive mapping formalism as well as corresponding reasoning algorithms for query answering.

The query processing follows a two-step process consisting of: (1) the selection of relevant resources based on metadata managed in a metadata registry, (2) query answering against relevant resources, which are integrated using a virtual ontology that logically imports relevant ontologies and mappings. This virtual integration provides global model semantics as if all ontologies were integrated locally. Practically, the distributed ontologies still reside on the remote nodes. The algorithm for question answering based on the correspondence between description logics and disjunctive datalog, which is implemented in KAON2, only requires the TBox part and the ABox predicate extensions that are actually relevant for the evaluation of the query to be accessed. Our evaluation results show that the approach is very promising as performance of query answering is essentially dominated by the size of the data, and only slightly affected by the degree of distribution and heterogeneity. In fact, the performance is comparable to settings where the data resides on a single, homogeneous node.

There are several directions of future work: As we currently assume that mappings between heterogeneous nodes already exist a priori, an obvious improvement would be the use of automated mapping tools for the online discovery of mappings between ontologies. Further, we consider the use of alternative mapping formalism [18] with different characteristics with respect to expressiveness, domain assumptions, and dealing with local inconsistencies. Finally, we will investigate other relations between networked ontologies, including modularization and version relationships, for which our existing reasoning algorithms need to be extended.

Acknowledgments

Research reported in this paper has been partially financed by the EU in the IST projects SEKT (IST-2003-506826, `http://www.sekt-project.com/`) and NeOn (IST-2006-027595, `http://www.neon-project.org/`). The KAONp2p has been implemented to a large extent by Rui Guo from EPFL Lausanne as part of his master's thesis at the University of Karlsruhe.

8. REFERENCES

[1] K. Aberer, P. Cudré-Mauroux, M. Hauswirth, and T. V. Pelt. Gridvine: Building internet-scale semantic overlay networks. In McIlraith et al. [15], pages 107–121.

[2] M. Arenas, V. Kantere, A. Kementsietsidis, I. Kiringa, R. J. Miller, and J. Mylopoulos. The hyperion project: From data integration to data coordination. *SIGMOD Record*, 32(3), 2003.

[3] D. Calvanese, G. D. Giacomo, and M. Lenzerini. A framework for ontology integration. In *Proceedings of the First Semantic Web Working Symposium*, pages 303–316, 2001.

[4] M. Ehrig, P. Haase, N. Stojanovic, and M. Hefke. Similarity for ontologies - a comprehensive framework. In *13th European Conference on Information Systems*, MAY 2005.

[5] E. Franconi, G. M. Kuper, A. Lopatenko, and I. Zaihrayeu. The coDB robust peer-to-peer database system. In M. Agosti, N. Dessì, and F. A. Schreiber, editors, *SEBD*, pages 382–393, 2004.

[6] Y. Guo, Z. Pan, and J. Heflin. Lubm: A benchmark for owl knowledge base systems. journal of web semantics. *Journal of Web Semantics*, 3(2):158–182, 2005.

[7] P. Haase, J. Broekstra, M. Ehrig, M. Menken, P. Mika, M. Plechawski, P. Pyszlak, B. Schnizler, R. Siebes, S. Staab, and C. Tempich. Bibster - a semantics-based bibliographic peer-to-peer system. In McIlraith et al. [15].

[8] P. Haase and B. Motik. A mapping system for the integration of owl-dl ontologies. In A. Hahn, S. Abels, and L. Haak, editors, *IHIS*, pages 9–16. ACM, 2005.

[9] P. Haase, R. Siebes, and F. van Harmelen. Peer selection in peer-to-peer networks with semantic topologies. In *Proceedings of the First International IFIP Conference on Semantics of a Networked World: ICSNW 2004, Paris, France, June 17-19, 2004.*, pages 108–125, 2004.

[10] J. Hartmann, Y. Sure, P. Haase, R. Palma, and M. C. Surez-Figueroa. OMV – ontology metadata vocabulary. In C. Welty, editor, *ISWC 2005 Workshop on Ontology Patterns for the Semantic Web*, NOV 2005.

[11] U. Hustadt, B. Motik, and U. Sattler. Reducing $\mathcal{SHIQ}^-$ Description Logic to Disjunctive Datalog Programs. In *Proceedings of the 9th Conference on Knowledge Representation and Reasoning (KR2004)*, pages 152–162. AAAI Press, June 2004.

[12] U. Hustadt, B. Motik, and U. Sattler. Data Complexity of Reasoning in Very Expressive Description Logics. In *Proceedings IJCAI 2005*, Edinburgh, UK, 2005. Morgan-Kaufmann.

[13] M. Lenzerini. Data integration: a theoretical perspective. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 233–246. ACM Press, 2002.

[14] A. Maedche, B. Motik, L. Stojanovic, R. Studer, and R. Volz. An infrastructure for searching, reusing and evolving distributed ontologies. In *Proceedings of the Twelfth International World Wide Web Conference (WWW 2003)*, pages 439–448. ACM, 2003.

[15] S. A. McIlraith, D. Plexousakis, and F. van Harmelen, editors. *The Semantic Web - ISWC 2004: Third International Semantic Web Conference,Hiroshima, Japan, November 7-11, 2004. Proceedings*, volume 3298 of *Lecture Notes in Computer Science*. Springer, 2004.

[16] B. Motik, U. Sattler, and R. Studer. Query answering for OWL-DL with rules. In McIlraith et al. [15], pages 549–563.

[17] R. Palma and P. Haase. Oyster - sharing and re-using ontologies in a peer-to-peer community. In Y. Gil, E. Motta, V. R. Benjamins, and M. A. Musen, editors, *International Semantic Web Conference*, volume 3729 of *LNCS*, pages 1059–1062. Springer, 2005.

[18] L. Serafini, H. Stuckenschmidt, and H. Wache. A formal investigation of mapping language for terminological knowledge. In L. P. Kaelbling and A. Saffiotti, editors, *IJCAI*, pages 576–581. Professional Book Center, 2005.

[19] L. Serafini and A. Tamilin. Drago: Distributed reasoning architecture for the semantic web. In *Proceedings of the Second European Semantic Web Conference, ESWC 2005, Heraklion, Crete, Greece, May 29 - June 1, 2005*, pages 361–376, 2005.

[20] E. Sirin and B. Parsia. Pellet: An OWL DL reasoner. In V. Haarslev and R. Möller, editors, *Description Logics*, volume 104 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2004.

[21] I. Tatarinov, Z. Ives, J. Madhavant, A. Halevy, D. Suciu, N. Dalvi, X. Dong, Y. Kadiyska, G. Miklau, and P. Mork. The piazza peer data management project. *SIGMOD Record*, 32(3), 2003.