Quality-oriented and Metadata-driven Integration in Information Grids

Christoph Quix

RWTH Aachen, Informatik V, 52056 Aachen, Germany,
quix@cs.rwth-aachen.de,
http://www-i5.informatik.rwth-aachen.de/~quix

Abstract. The goal of information grids is to provide a virtually integrated view on information, which is physically stored in many distributed nodes of the grid. A user should be able to query the grid through a uniform query interface using a common data model, without knowing the details of the distribution of the data. Information grids that integrate information from heterogeneous resources have to resolve the problems of semantic and structural heterogeneity, and also take into account different quality characteristics of the sources. This paper addresses the integration in information grids, and adopts results for data integration in data warehouses to the information grid. The contributions of this paper are (i) an extended metadata framework to capture the different types of metadata of an information grid, (ii) the integration of quality aspects into this framework, and (iii) a methodology for a quality-oriented and metadata-driven integration of information.

1 Introduction

The Grid has been defined as a system that "coordinates resources that are not subject to centralized control" and "delivers nontrivial qualities of service" [10]. In addition to computational grids that provide a high computational performance by combining and sharing several high-end computing devices, there are many approaches to use the grid architecture to share data and information [7,15]. The goal of these grids is to provide a virtually integrated collection of information, which is physically stored in many distributed nodes of the grid. A user should be able to query the grid through a uniform query interface using a common data model, without knowing the details of the distribution of the data. We call such grids in the following *information grids* [15].

The additional qualities of service provided by information grids are an improved query performance by making use of the parallel architecture, and/or the semantic integration of heterogeneous information. The first point is addressed in particular in research projects on managing large data collections in physics or life sciences [1,11,12]. In these contexts, the data in the nodes have a homogeneous strucuture, i.e. schemas and query processing capabilities in each node are similar. The main challenge is the efficient parallel processing of the queries and the combination of the results.

In this paper, we will focus on the second type of information grids, which integrate information from a set of *heterogeneous*, *independent*, and *dynamic* data resources[1]. Heterogeneous means that the structure and the semantics of the various data resources may be very different, e.g. the resources use XML and relational models to structure the data, or they do not have a common terminology to define the schemas. The resources are independent, because they are not under a central control. Dynamic means, that the resources might change their schema, or they are added or removed from the grid at anytime.

Although there have been many approaches to data integration in different areas, the problem has not been solved in general. Data integration requires still a lot of manual effort, for example, to define the mappings between the data sources and the integrated data model, or to resolve data quality problems. To overcome these problems, frameworks have been defined that should simplify the integration process, e.g. in data warehouses [5,16]. The integration of heterogeneous information from independent and dynamic sources is also addressed in the semantic web [9,18]. In addition to the problem of semantic integration of heterogeneous information, a successful integration requires also a technological infrastructure that enables the communication and interaction of the various components involved. As such an infrastructure is missing in general in the semantic web, research is now addressing the combination of technologies from the semantic web, the grid, and web services [13,28]. It has also been shown that data quality is an important issue in integrated environments [30]. The usage of the data in a different context than originally planned poses new requirements to the data, which might result in quality problems.

This paper addresses the integration in information grids, and adopts approaches from other areas to the information grid. The contributions of this paper are (i) an extended metadata framework to capture the different types of metadata of an information grid, (ii) the integration of quality aspects into this framework, and (iii) a methodology for a quality-oriented and metadata-driven integration of information. Some of the points presented in this paper may sound like *old wine in new bottles*. This is partly true, as this work is based on our previous work in the context of data warehouses. But the basic question that has to be addressed in the context of information grids is: which techniques, that have been developed in the area of database management, data integration, or data warehouses, can be re-used in information grids? And if a technique can be re-used, what kind of adaptations are necessary to use the technique in a grid environment.

The paper is structured as follows. In section 2, we will first present the methodology for quality-oriented and metadata-driven integration of information. Section 3 gives then an overview of the metadata model of the repository, which delivers the required information for the integration process. In section 4, we will present a meta model that represents the quality information required for the quality-oriented integration methodology.

[1] In this paper, we adopt the terminology of the OGSA-DAI framework [24]. A data resource is a node (or peer) in the grid, which provides some kind of data.

2 A Methodology for Integration in Information Grids

In this section, we present a methodology for the quality-oriented and metadata-driven integration in information grids. Input to this methodology is an information request of a user, expressed as a query in terms of an ontology. Taking into account the mappings between this ontology and the schemas of the data resources, the user query should be translated into separate queries to the data resources, and the results of the resources have to be combined later to form a uniform result.

As the information grid might contain some redundancy, i.e. the same information is stored in various data resources with different quality characteristics, there might be not one unique solution to answer the query. Therefore, the quality requirements of the user should be taken into account to select the "best" solution.

This scenario is not new, it is known in a more static form in data warehouses [5] (where the queries to the data (re)sources are implemented as extraction programs), or in a more dynamic, loosely-coupled form in Peer-to-Peer systems [2,29] (where no central ontology is available). We adopt here a methodology that has been developed in the context of data warehouses [27]. The methodology combines several approaches for data integration and quality management (e.g. [5,23,26]) into a holistic approach to quality-oriented data integration.

The main difference to data integration in data warehouses is that we do not assume that the result of the query has to be complete, i.e. include all possible answers. Especially in the context of the World Wide Web, computing the complete answer is not always necessary and feasible [22]: users are often interested only in the top ten results of a search engine, or the computation of the complete result is too expensive in terms of processing time or monetary costs (data sources may cost money). Therefore, our approach is not to query by default all possible data sources, but only those that will deliver the "best" data quality. The term "best" is quoted in this context, because whatever is the best result of a query depends on the requirements of the user.

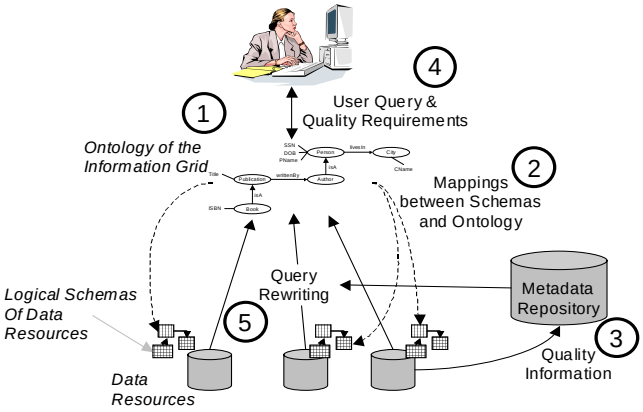

Fig. 1. Methodology for Integration in Information Grids

An overview of the methodology is given figure 1. In the following subsections, we will address briefly each phase of the methodology (according to the numbers in the figure) in more detail. Due to lack of space, we cannot give the formal definition of the whole methodology, but we will explain the most important concepts using a running example.

2.1 Ontology of the Information Grid

A prerequisite for the methodology is the availability of an ontology that defines the semantics of the information in the grid. To define this ontology, an ontology standard such as OWL or DAML can be used. We do not restrict the ontology language to a specific variant, but it should be possible to define classes with properties and relationships between these classes (such as *equivalentClass* or *subClassOf*). As we will see later, it is important that we are able to prove whether two classes of the ontology are sub-classes of each other, have a non-empty intersection, or are not related at all.

It is known that the development of such a "global" ontology can be a hard task. But within this methodology, the development of the ontology is made simpler by allowing the ontology and the data resources to evolve independently from each other. Furthermore, if the information grid is established for a particular "virtual" organization, e.g. a specific industry sector, an ontology might be already available.

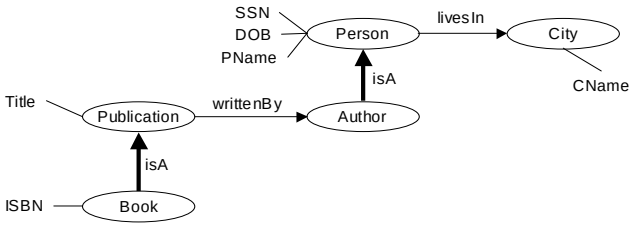

Fig. 2. Example Ontology with Publications, Books, and Authors

An example of an ontology that we will use in the following is shown in figure 2. *Publications* are written by *Authors*; *Book* is a sub-class of *Publication* with an additional property ISBN; *Authors* are *Persons* with properties SSN, date of birth (DOB), and a name; and *Persons* are living in a *City*.

2.2 Mappings between Schemas and Ontology

Each data resource in the information grid has to provide a schema, so that it is known how the information is structured. The logical schemas of the resources is defined as a view on the ontology. In data integration, this approach is known as local-as-view [14]. As aforementioned, this approach has the advantage that the ontology is independent of the data resources (in contrast to the global-as-view approach, where the definition of

the ontology depends on the data resources). This means that a change in a data resource does not require a change in the ontology.

A basic problem in this context is the mapping between logical schemas and conceptual ontologies. A logical schema describes the structure of actual data (e.g. a person is tuple consisting of an ID, a SSN, a name, and a date), whereas the ontologies describe "abstract" objects (e.g. a person is an object that has a name, a SSN, and a date of birth) without going into detail about the data format. To bridge this gap, we follow here the approach proposed in [5]: the elements of the logical schema are defined as annotated queries over the ontology. The query is a conjunctive query, expressed in a form similar to Datalog. The annotation of the query defines the relationships between the logical data values and the conceptual objects.

In our example, we define data resources containing information about persons, books, and authors. Resource 1 has three relations: [2]

$$\begin{aligned}
\mathsf{Books}_1(isbn, title) &\leftarrow \mathsf{Book}(b), \mathsf{ISBN}(b, i), \mathsf{Title}(b, t) \\
&\quad | \ \mathsf{identify}(isbn, b), \mathsf{identify}(isbn, i), \mathsf{identify}(title, t) \\
\mathsf{WrittenBy}_1(isbn, aid) &\leftarrow \mathsf{Book}(b), \mathsf{ISBN}(b, i), \mathsf{WrittenBy}(b, a), \mathsf{Author}(a) \\
&\quad | \ \mathsf{identify}(isbn, b), \mathsf{identify}(isbn, i), \mathsf{identify}(aid, a) \\
\mathsf{Author}_1(aid, name, dob) &\leftarrow \mathsf{Author}(a), \mathsf{PName}(a, n), \mathsf{DOB}(a, d) \\
&\quad | \ \mathsf{identify}(aid, a), \mathsf{identify}(name, n), \mathsf{identify}(dob, d)
\end{aligned}$$

The relation Books_1 has two columns, ISBN and Title. The first line defines the query over the ontology: Book, ISBN, and Title refer to elements in the ontology. The second line after the bar defines the relationship between the conceptual objects (represented by the variables b, i, and t), and the relational data (represented by the variables $isbn$ and $title$). The predicate identify relates a conceptual variable to a relational variable. For example, $\mathsf{identify}(isbn, b)$ means that conceptual objects represented by b can be identified by the value $isbn$. The second resource has information about persons and their place of living:

$$\begin{aligned}
\mathsf{Person}_2 \quad & (ssn, name, dob, city) \leftarrow \\
& \mathsf{Person}(p), \mathsf{PName}(p, n), \mathsf{DOB}(p, d), \mathsf{LivesIn}(p, c), \mathsf{CName}(c, cn) \\
& | \ \mathsf{identify}(ssn, p), \mathsf{identify}(name, n), \mathsf{identify}(dob, d), \mathsf{identify}(city, cn)
\end{aligned}$$

The third resource is similar to the first one, except that the author table has an additional attribute "city" that represents the place of living of the author.

$$\begin{aligned}
\mathsf{Author}_3(aid, name, dob, city) &\leftarrow \mathsf{Author}(a), \mathsf{PName}(a, n), \mathsf{DOB}(a, d), \mathsf{City}(a, c) \\
&\quad | \ \mathsf{identify}(aid, a), \mathsf{identify}(name, n), \mathsf{identify}(dob, d), \\
&\quad \ \ \mathsf{identify}(city, c)
\end{aligned}$$

[2] For the purpose of this paper, we assume that the schemas have a flat relational structure without nested sub-elements. Nested structures of semi-structured or object-oriented data models could be represented using canonical mapping to flat structures, although we know that this is only a sub-optimal solution.

2.3 Metadata Repository and Quality Information

The metadata repository manages all metadata relevant for the task of information integration. In particular, it contains information about the relationships between ontologies and logical schemas as defined in the previous section, and also quality information. The quality information represents some quality measures of the data resources such as availability, responsiveness, or correctness. Based on these quality measures and the requirements of a user, a ranking of possible query plans will be done in the query rewriting phase. More details about the metadata model and the quality information will be given in sections 3 and 4.

2.4 User Query and Quality Requirements

The query of the user is defined as a conjunctive query over the ontology of the information grid. The definition is done in a similar way as the definition of the logical schema, i.e. the definition consists of a query and an annotation. For example, the following query retrieves authors living in Munich with name and titles of their books:

$$\begin{aligned} \mathsf{Q}(name, title) \leftarrow\ & \mathsf{Book}(b), \mathsf{Author}(a), \mathsf{WrittenBy}(b, a), \mathsf{PName}(a, n), \mathsf{Title}(b, t), \\ & \mathsf{LivesIn}(a, c), \mathsf{CName}(c, cn), cn =' Munich' \\ \mid\ & \mathsf{identify}(name, n), \mathsf{identify}(title, t) \end{aligned}$$

The quality requirements of the user are represented in two ways. First, the user can attach a weight (expressed as a value between 0 and 1) to each predicate of the query that refers to the ontology. Using this technique, the user can state that the certain constraints of the query are more important than others. For example, the user could state that the information about the relationship WrittenBy is more important than the relationship LivesIn. In addition, the user can attach a weight to each quality dimension. Thereby, the user can state that some quality dimensions are more important than others, e.g. response time is more important than completeness. For each possible query plan, a quality value is calculated by using specific merge functions (e.g. to compute an estimated quality value for the result of a join operation). The Simple Additive Weighting method is used to combine quality values of different quality dimensions [21,27].

2.5 Query Rewriting

To rewrite the user query into several queries to the data resources, first we have to identify data resources, which are relevant to the query. A data resource is relevant, if it is able to contribute to the result of the query. This can be formally proven by showing that the intersection of a predicate of the query and a predicate of the definition of the logical schema is non-empty. In our example, we can easily verify that all resources are relevant as they use the same predicates as the user query (e.g. $\mathsf{Book}(b)$).

Now, as we have identified the relevant resources, we have to find all possible combinations of resources that result in a "correct" rewriting of the original query. A rewriting is considered to be "correct" if it delivers only results that fulfill the constraints of the original query. Note that this does not mean that the rewriting is "complete" or

"equivalent" to the original query, i.e. it is not guaranteed that all results will be found. To find the rewriting, we have adopted the *MiniCon* algorithm [26], which is an efficient algorithm to find correct rewritings. However, the correctness of the query rewriting is only assured, if the resources deliver a subset of the relevant answers of the query. To guarantee correct rewritings, we would have to consider only those data resources as relevant for which the predicate in the definition of the logical schema describes a subset of the corresponding predicate of the query. As mentioned before, we think that this restriction is too hard in the context of information grids as it would rule out many possible query plans. As solution to this problem, we propose to attach additional quality factors to a query plan, which indicate the correctness and relevance of the query plan. The "best" query plans are then selected according to the quality requirements of the user (e.g. complete answer is more important than a fully correct answer).

The basic MiniCon algorithm works in two phases. In the first phase, the predicates of the user query are mapped on the predicates of the definition of a relation. The result is stored in a so-called MiniCon-Description (MCD). The MCD consists in particular of a mapping of the variables between the queries and a list of predicates of the query that are covered by a data resource. In our example, consider the first relation Books_1 in resource 1. It contains the predicates $\mathsf{Book}(b)$ and $\mathsf{Title}(b, t)$, which are also contained in the user query. Thus, these two predicates are *covered* by the relation Books_1. As another example, consider the definition of the relation Person_2. It contains a predicate $\mathsf{Person}(p)$, which does not appear in the user query, but the predicate $\mathsf{Author}(a)$ has been declared to be a subclass of Person, so the relation is relevant to the query. In addition, this relation covers the predicates $\mathsf{PName}(a, n)$, $\mathsf{LivesIn}(a, c)$, and $\mathsf{CName}(c, cn)$.

In the second phase of the MiniCon algorithm, the MCDs are combined in such a way that all predicates of the user query are covered. In our example, this results in several possible query plans. The obvious plans are a join between the relations of resource 1 and 2, or a separate query to resource 3. But it is also possible to combine the resources in a different way, e.g. take the author information from resource 3, the information about books from resource 1, and the information about the place of residence from resource 2. As it is very likely that this solution has a lower quality than the previous solutions, it will probably not get a high quality value based on the computation sketched in the previous subsection.

We have extended the MiniCon algorithm with a third phase, which also takes the annotation of the queries into account. Remember that the annotation provided the relationship between the relational and conceptual variables. So far, we have considered only the conceptual variables. But we have also to take into account the relational variables as these variables are necessary to express the join conditions between the relations. In the example, consider the combination of the relations Author_1 and Person_2. We cannot perform the join directly between these relations, as they do not have common fields. Therefore, we have to create a match function that matches an entry of the relation Author with an entry of the relation Person, may be using the fields name and DOB. As the queries in this environment will be defined in an ad-hoc manner, the system has to provide some domain-specific match functions (e.g. to match names or addresses), complemented by some general match functions (e.g. based on record linkage techniques such as [17]).

3 Metadata Model

To enable an efficient integration of data in a distributed environment, the management of metadata is necessary. In the context of internet information systems several metadata standards have already been developed. For example, the *Web Ontology Language* (OWL) [20] can be used to describe the semantical relationships of concepts in the system, whereas the *Web Services Description Language* (WSDL) [8] can be used the describe the technical properties of a web service. However, these standards are more or less isolated, they do not represent the relationships between the different perspectives.

We propose an integrated metadata model which links the metadata from the different perspectives of an information system. According to the classical separation in database systems and our previous work [16], we classify the metadata of a resource in the information grid into a *conceptual*, *logical* and *physical* perspective (see also figure 3):

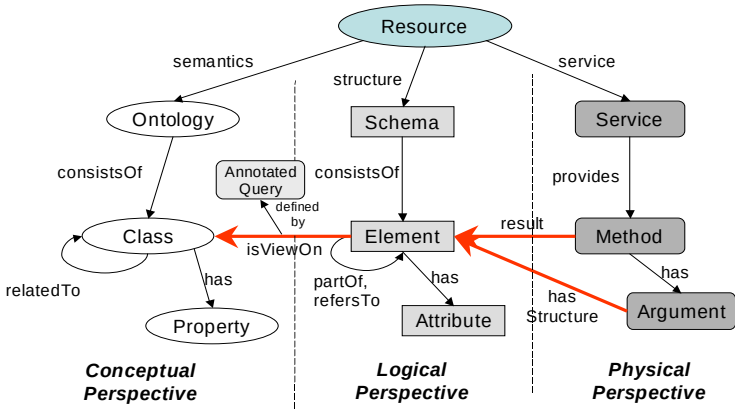

Fig. 3. Overview of the Meta Model for Information Grids

– The conceptual perspective represents the semantical models of the information grid, i.e. the ontologies. The objects in this perspective should be described using an existing standard such as OWL. Although the model allows the definition of individual ontologies for each resource in the information grid, there has to be an integrated ontology that unifies all ontologies of the information grid. This integrated ontology will be presented also to the user to enable the formulation of queries. Each data resource could extend this integrated ontology with resource-specific concepts if necessary.
– The logical perspective models the structure of the data of the individual resources of the information grid. As described in section 2.2, the logical schema will be defined as view on the conceptual schema. The horizontal link *isViewOn* between *Element* and *Class* represents this relationship and has also a link to the formal definition of this view as annotated query.

– The physical perspective describes where the data is located and how it can be accessed. This is done by specifying the type of service of the resource (e.g. SOAP, OGSA-DAI [24], or ODBC). Again, the important information is represented by the horizontal link to the logical perspective, which provides the information what kind of schema is used in the result, and which arguments are required for a method provided by the service.
– The object *Resource* joins the metadata of one data resource together.

The combination of metadata from different aspects has also been proposed in the SWAP project [3]. Broekstra et al. present a metadata model for peer-to-peer systems that combines ontological structures with information needed for query processing. However, their approach does not make a clear distinction between the different types of metadata. It also includes some quality information of the peer, such as trust and confidence. Other approaches for metadata management in grids focus mainly only the logical and physical metadata of data resources (e.g. [6,19]).

4 Data Quality in Information Grids

In short, data quality can be defined as "fitness for use" [30]. This means that the quality of data cannot be measured without taking the usage context into account. This is especially important in integrated information systems, in which data is used in many different contexts with different quality requirements. The discrepance of the requirements for data sources and data warehouses is also a big hurdle in the design and implementation of the data integration process in data warehouses and has lead to many failures of data warehouse projects.

In the context of information grids, the consideration of data quality is also necessary as the requirements that have been used to develop the data resources might be very different from the requirements of a user of the information grid. Furthermore, the goal of an information grid should be to deliver a query result as good as possible. Therefore, we have to know how we can characterize a good result. Finally, the query processing can be simplified if we can restrict the queries to data resources that the deliver the required quality.

In the context of data warehouses, we proposed a quality model [16] based on the Goal-Question-Metric approach (GQM) [25]. The model is well suited for data warehouses, as it allows the administrators to monitor the system with respect to quality goals of users. The main feature of the model is the separation between the quality goals of users on the one hand, and the measurements of quality factors on the other hand. This separation is motivated by the observation that we cannot measure the rather abstract and high-level quality goals of users directly. Instead, we measure quality factors, which represent some (quality) properties of an object in the data warehouse, such as response time or number of system crashes in a month. To bridge the gap between quality goals and factors, the GQM-approach uses questions, which evaluate quality factors and provide an evidence for the fulfillment of a quality goal. For example, a quality goal like *"Improve the availability of system A"* could have questions like *"What is the response time for queries?"* or *"How many crashes had the system last month?"*, which could be answered by using the quality factors mentioned before.

The GQM-approach is especially useful for software quality management as the step-by-step refinement from quality goals to measurements helps to identify quality factors that should be measured for a system (we can not measure everything as the measurements require also some effort).

However, in information grids, we do not have a central administrator who can take care of quality goals of individual users, as the grid is a more open, more flexible, and more dynamic system than a data warehouse. In this context, the main task of the quality model is to deliver the quality values, which are required for the integration methodology described in section 2. Therefore, we propose a simplified quality model in which the quality factors are the central component (see figure 4).

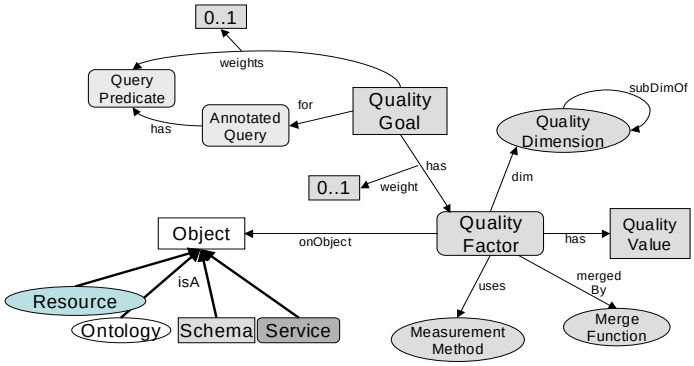

Fig. 4. Meta Model to represent Quality Information

Quality Factors represent measurements, executed by a specific *Measurement Method* that delivers a *Quality Value* as result. Quality factors are related to a quality dimensions that are classified by a hierarchical structure. They measure quality properties of some *Object* of the information grid, which can be either an object from the conceptual, logical, or physical perspective. Quality goals are defined by the user for annotated queries that are used to query the system. As described in section 2, the quality requirements are specified by weighting the quality factors and the query predicates. In addition, we need to know what kind of merging function can be used for a quality factor, if results are joined from several resources. A merging function for the quality factor "response time" could be MAX, as we can do the querying of the resources in parallel.

It is still possible to extend this quality model to the full GQM-model presented in [16], if goals and questions become more important, e.g. during the design phase of the information grid, or if quality factors need to be identified.

5 Conclusion

Information grids are a new architecture for the integration of heterogeneous information. Although they provide a flexible and powerful technological infrastructure, the information integration in grids faces similar problems concerning semantical and structural heterogeneity as in data warehouses or the semantic web.

In this paper, we have proposed an integrated methodology for the quality-oriented and metadata-driven integration in information grids. The approach is based on our previous experience in the integration of information systems and adopts techniques developed in the context of data warehouses to the information grid. In particular, we have proposed a metadata framework for the management of metadata in an information grid. Furthermore, we presented a quality model, which enables the representation of quality goals and measurements in the information grid.

Future work will investigate the integration of a service-oriented view in this context, going into the direction of semantic web services [4]. Delivering the result of a query can be also seen as service provided by some web resource. We also plan a prototypical implementation of the current methodology based on the OGSA-DAI framework [24]. To enable an effective quality management, quality dimensions and factors for information grids have to be defined, as we have done it in our previous work for data warehouses [16,27].

Acknowledgements: This work is supported in part by the 5th Framework IST programme of the European Communities through the project SEWASIE (IST-2001-34825, `http://www.sewasie.org`).

References

1. P. Avery, I. Foster. The GriPhyN Project: Towards Petascale Virtual Data Grids, 2001. `http://www.griphyn.org`.
2. P. A. Bernstein, F. Giunchiglia, A. Kementsietsidis, J. Mylopoulos, L. Serafini, I. Zaihrayeu. Data Management for Peer-to-Peer Computing : A Vision. In *5th Intl. Workshop on the Web and Databases (WebDB)*, pp. 89–94. Madison, WI, 2002.
3. J. Broekstra, M. Ehrig, P. Haase, F. van Harmelen, A. Kampman, M. Sabou, R. Siebes, S. Staab, H. Stuckenschmidt, C. Tempich. A Metadata Model for Semantics-Based Peer-to-Peer Systems. In *Proc. WWW'03 Workshop on Semantics in Peer-to-Peer and Grid Computing*. 2003.
4. C. Bussler. Semantic Web Services: The Future of Integration! In *7th East European Conference on Advances in Databases and Information Systems (ADBIS)*, *LNCS*, vol. 2798, pp. 1–2. Springer-Verlag, Dresden, Germany, 2003.
5. D. Calvanese, G. D. Giacomo, M. Lenzerini, D. Nardi, R. Rosati. Data Integration in Data Warehousing. *International Journal of Cooperative Information Systems (IJCIS)*, **10**(3):237–271, 2001.
6. M. Cannataro, C. Comito, A. Congiusta, C. Mastroianni, A. Pugliese, D. Talia, P. Trunfio, P. Veltri. Architecture, Metadata and Ontologies in the Knowledge Grid. In *IST Workshop on Metadata Management in Grid and P2P Systems*. London, UK, 2003.
7. A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, S. Tuecke. The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Data Sets. *Journal of Network and Computer Applications: Special Issue on Network-Based Storage Services*, **23**(3):187–200, 2000.
8. R. Chinnici, M. Gudgin, J.-J. Moreau, J. Schlimmer, S. Weerawarana. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. *W3C Working Draft*, World Wide Web Consortium, Aug 2004. `http://www.w3.org/TR/wsdl20/`.
9. V. Christophides, G. Karvounarakis, A. Magkanaraki, D. Plexousakis, V. Tannen. The ICS-FORTH Semantic Web Integration Middleware (SWIM). *IEEE Data Engineering Bulletin*, **26**(4):11–18, 2003.

10. I. Foster. What is the Grid? A Three Point Checklist. *Grid Today*, **1**(6), July 22, 2002. http://www.gridtoday.com/02/0722/100136.html.
11. I. Foster, J.-S. Voeckler, M. Wilde, Y. Zhao. The Virtual Data Grid: A New Model and Architecture for Data-Intensive Collaboration. In *First Biennial Conference on Innovative Data Systems Research (CIDR 2003)*. Asilomar, CA, USA, 2003.
12. J. Geijer, B. Lenhard, R. Merino-Martinez, G. Norstedt, A. Flores-Morales. Grid Computing For The Analysis Of Regulatory Elements In Co-Regulated Sets Of Genes. *Parallel Processing Letters*, **14**(2):137–150, 2004.
13. C. A. Goble, D. de Roure. The Semantic Grid: Myth Busting and Bridge Building. In *Proceedings of the 16th Eureopean Conference on Artificial Intelligence (ECAI 2004)*, pp. 1129–1135. Valencia, Spain, 2004.
14. A. Y. Halevy. Answering queries using views: A survey. *VLDB Journal*, **10**(4):270–294, 2001.
15. M. Hyatt, R. Vrablik. The Information Grid – Secure access to any information, anywhere, over any network. *Tech. rep.*, IBM developerWorks, 2004. http://www-106.ibm.com/developerworks/library/gr-infogrid.html.
16. M. Jarke, M. A. Jeusfeld, C. Quix, P. Vassiliadis. Architecture and Quality in Data Warehouses: An Extended Repository Approach. *Information Systems*, **24**(3):229–253, 1999.
17. L. Jin, C. Li, S. Mehrotra. Efficient Record Linkage in Large Data Sets. In *8th Intl. Conference on Database Systems for Advanced Applications (DASFAA '03)*, pp. 137–146. IEEE Computer Society, Kyoto, Japan, 2003.
18. L. V. S. Lakshmanan, F. Sadri. Information Integration and the Semantic Web. *IEEE Data Engineering Bulletin*, **26**(4):19–25, 2003.
19. G. McCance. Metadata Management in the EU DataGrid. In *IST Workshop on Metadata Management in Grid and P2P Systems*. London, UK, 2003.
20. D. L. McGuinness, F. van Harmelen. OWL Web Ontology Language: Overview. *W3C Recommendation*, World Wide Web Consortium, Feb 2004. http://www.w3.org/TR/owl-features/.
21. F. Naumann. *Quality-Driven Query Answering for Integrated Information Systems*. Ph.D. thesis, Humboldt Universität zu Berlin, Springer-Verlag, 2002.
22. F. Naumann, J.-C. Freytag, U. Leser. Completeness of integrated information sources. *Information Systems*, **29**(7):583–615, 2004.
23. F. Naumann, U. Leser, J. C. Freytag. Quality-driven Integration of Heterogenous Information Systems. In *25th Intl. Conference on Very Large Data Bases (VLDB)*, pp. 447–458. Edinburgh, Scotland, 1999.
24. OGSA-DAI. Open Grid Services Architecture – Data Access and Integration, 2004. http://www.ogsadai.org.uk/.
25. M. Oivo, V. Basili. Representing Software Engineering Models: The TAME Goal Oriented Approach. *IEEE Trans. on Software Engineering*, **18**(10):886–898, 1992.
26. R. Pottinger, A. Y. Halevy. MiniCon: A scalable algorithm for answering queries using views. *VLDB Journal*, **10**(2-3):182–198, 2001.
27. C. Quix. *Metadata Management for quality-oriented Information Logistics in Data Warehouse Systems (in German)*. Ph.D. thesis, RWTH Aachen, Germany, 2003.
28. D. de Roure, N. R. Jennings, N. Shadbolt. The Semantic Grid: A Future e-Science Infrastructure. *Int. J. of Concurrency and Computation: Practice and Experience*, **15**(11), 2003.
29. I. Tatarinov, Z. G. Ives, J. Madhavan, A. Y. Halevy, D. Suciu, N. N. Dalvi, X. Dong, Y. Kadiyska, G. Miklau, P. Mork. The Piazza peer data management project. *SIGMOD Record*, **32**(3):47–52, 2003.
30. G. K. Tayi, D. P. Ballou. Examining Data Quality. *Communications of the ACM*, **41**(2):54–57, Feb 1998.