Towards Large-Scale Scientific Dataspaces for e-Science Applications

Ibrahim Elsayed and Peter Brezany

University of Vienna, Department of Scientific Computing
Nordbergstrasse 15/C/3, A-1090 Vienna, Austria
{elsayed,brezany}@par.univie.ac.at
http://www.par.univie.ac.at/

Abstract. This work intends to provide a large-scale scientific data management solution based on the concepts of dataspaces for e-Science applications. Our approach is to semantically enrich the existing relationship among primary and derived data items, and to preserve both relationships and data together within a dataspace to be reused by owners and others. To enable reuse, data must be well preserved. Preservation of scientific data can best be established if the full life cycle of data is addressed. This is challenged by the e-Science life cycle ontology, whose major goal is to trace semantics about procedures in scientific experiments. jSpace, a first prototype of a scientific dataspace support platform is implemented and deployed to an early core of adopters in the breath gas research domain from which specific use cases are derived. In this paper we describe the architecture, discuss a specific prototype implementation and outline the design concepts of a second prototype.

Keywords: Scientific Dataspace, Scientific Data Management, e-Science.

1 Introduction

Dataspaces are not a data integration approach, rather they are a data coexistence approach [1]. The goal is to rise the abstraction level at which data is managed. Dataspaces consist of participants and relationships. Participants can be any data element and relationships should be able to model any interconnection among these participants. Dataspace support platforms (DSSPs) represent the collection of software pieces and services that control the organization, storage and retrieval of data in a dataspace. The challenges of dataspaces discussed in [2] have influenced many research groups of the data management community. However, most effort was put on the mainstream related dataspace research [3,4,5] or on development of personal dataspace systems [6,7].

A major challenge faced by the scientific data management community is to efficiently organize data products used in and generated by scientific experiments of diverse e-Science applications. The challenge to derivate history of a data product is known as data provenance [8]. In contrast to the mainstream related dataspace research scientific dataspaces can be seen as an umbrella to those

M. Yoshikawa et al. (Eds.): DASFAA 2010, LNCS 6193, pp. 69–80, 2010.

research challenges with the goal to establish a distributed large-scale scientific repository where the full life cycle of scientific data is well preserved. In our previous work we have addressed a scientific dataspace paradigm [9] aiming at semi-autonomous creation of semantically rich relationships among data sets used in scientific studies and further, at preserving both, relationships and their corresponding data sets within a distributed space of data.

Breath gas analysis in medicine [10], an emerging new scientific field with a growing international scientific community, represents our driving life science application from which we derive specific dataspace use cases [11]. Breath gas researcher are addressing many different breath gas studies in terms of investigating and screening for hundreds of compounds in exhaled breath gas. Such studies include several experiments. There is a need to provide structured representations of scientific experiments for the breath gas analysis domain. It is hardly possible for a breath gas researcher to re-run an experiment that was conducted at a different research lab from his own, far less to understand its semantics. Both formal research works in publications and also scientific data used in or produced by corresponding experiments should be interlinked with semantics. The e-Science life cycle ontology [12] addresses the precise description of scientific experiments by taking advantage of well-defined semantics of the Resource Description Framework (RDF) [13] and the expressive formal logic-based OWL language [14]. Experiments described by the ontology are referred to as *Life Cycle Resources* (LCRs). A LCR in fact represents the semantic relationship among dataspace participants. We differentiate three kinds of participants (a) data sources being accessed for investigation (primary data), (b) its corresponding findings (derived data), and (c) the set of activities defining concrete preprocessing and analysis methods (background data). Instances of those kinds of data sets are interconnected by LCRs within the scientific dataspace.

In this paper we discuss the implementation of *jSpace*, which is our first experimental prototype of a scientific dataspace system based on semantic web technologies. It represents a further development of the dataspace paradigm introduced in [1], in particular it implements the scientific dataspace framework proposed in [9]. jSpace is an important key-point for collaboration of members of the breath gas analysis research community. After a brief review of related work in dataspace research in the next section we describe the architecture of the scientific dataspace support platform by breaking the system into its major tools in Section 3. That followed, we present the implementation status of jSpace version 1.0 in Section 4 and discuss it in Section 5. Finally, we conclude the paper and outline our next steps, which are towards developing the *semantic grid-enabled* version of jSpace in Section 6.

2 Related Systems

The concept of dataspaces introduced by Franklin et al. [1] gave rise to new data management challenges. Much dataspace research is applied in terms of personal information management. For example Yukun et al. describe in [7] a personal

dataspace management system, named OrientSpace, which implements data integration and data query functions. They introduce the CoreSpace framework, which represents a subspace of the personal dataspace containing only objects that are frequently accessed by the owner. The data model used is based on the vertical data model, which takes a vector to describe attributes of an object.

iMeMex [6] provides data management functionalities, such as querying, updating, performing backup and recovery operations. All data is presented using a single graph data model and queried using an own query language, called the iMeMex Query Language (iQL) [15]. Special converters convert the contents of data sources into the internal graph structure. Core idea of iMeMex is a logical layer called Resource View Layer that abstracts from underlying subsystems and data sources. A Data Source Proxy connects to the data sources and provides plugins for file systems, IMAP email servers and RSS feeds, which shows that iMeMex is designed for personal information management, however not limited to it. Another personal dataspace management approach is proposed by Lei et al. in [16]. It introduces the Galaxy data model, which is an extension of the iMeMex data model in order to better consider security issues, primarily access policies.

Also the initiators of dataspaces proposed solutions to major dataspace research challenges like indexing dataspaces [3] and pay-as-you-go data integration approaches for dataspace systems [4,5]. This relates to the mainstream dataspace research. However, so far to our best knowledge no effort was put on applying dataspace concepts to e-Science applications in order to establish a large-scale scientific repository, that preserves scientific studies in conjunction with all its interconnected scientific data sets.

3 jSpace Architecture

In this Section we provide a summary of the architecture of jSpace. Main entities of the architecture (Figure 1) are the *Life Cycle Composer* - for creation of LCRs, the *RDF Store* - for storing those resources, the scientific dataspace itself - for storing participating data sets, the *Dataspace Indexer* - for their subscription, the *Search&Query Processor* allowing scientists to find those LCRs, and the *Dataspace Browser* for exploration of the dataspace. These, with each other cooperating software programs represent the environment in which the scientific dataspace is able to grow and evolve into a remarkable space of well preserved scientific data. They also provide the organization and retrieval of scientific data including their well defined semantics within the dataspace.

3.1 e-Science Life Cycle Composer

The aim of the e-Science life cycle composer is to provide a simple interface to the acting scientists to describe their experiments according to predefined attributes given by a scientific community. It can be seen as the feeding interface to the scientific dataspace. It is an easy but efficient way to capture semantically rich information about dataspace participants and relationships. It guides the

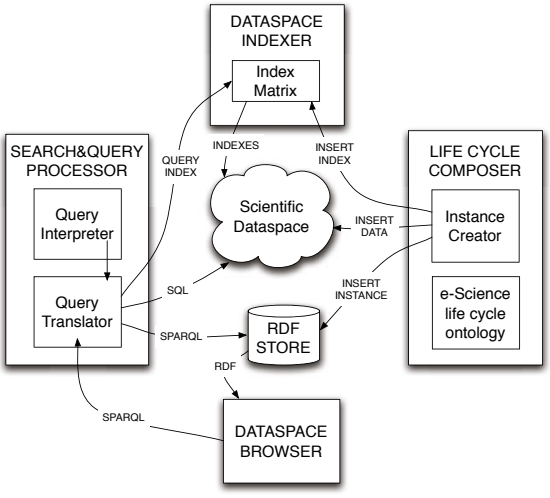

Fig. 1. Overview of the system architecture

user through the five e-Science life cycle activities, creates new individuals, and attaches them to a new LCR. It communicates with the *Dataspace Indexer*, which indexes new individuals. The indexing mechanism and its purpose are described in Section 3.4. Based on guidelines defined by responsible persons of the e-Science application domain to whom the dataspace is deployed, the scientist fills out a number of mandatory fields during experiment execution. Also references to data sets being used in the experiment are recorded. The information entered is used to create individuals of classes defined by the e-Science life cycle ontology. These individuals, consolidated within a LCR describe on a semantically high level a scientific experiment. The resulting RDF graph represents a LCR. It is saved within the RDF store.

3.2 RDF Store

The RDF store manages LCRs persistently. The SPARQL query language [17], which has been accepted as a W3C recommendation for querying RDF resources is used to query LCRs. There might be many dataspace instances set up at multiple research centers, which work together in terms of collaborative science. In such a very common scenario each center will host their own RDF store for storing their LCRs, which results in a distributed RDF data environment. There are two main approaches to handle the problem with multiple RDF stores.

Approach 1 - Global centralized RDF Store. This data warehouse alike approach provides a global centralized RDF store that organizes LCRs on a multi-institutional level. Local stores should guarantee high performance for the people working on the local site. Scientific experiments being conducted at any research lab that participates in a dataspace environment are stored in local stores as

long as access should be limited to researchers of the local organization. Once researchers want to share their experiments with other external collaborators or make them public to the scientific community, its corresponding LCRs will be stored in a central global store, which is shared with other dataspace instances.

Approach 2 - Distributed RDF Storage. This solution requires a middleware that supports federated SPARQL query processing. Currently, concepts from traditional approaches of federated query processing systems are adapted to provide integrated access to RDF data sources. Basic idea is to query a mediator, which distributes subqueries to local RDF stores and integrates the results. The DARQ engine [18] is an extension of the Jena-embedded query engine ARQ to support federated SPARQL queries. Very similar to the DARQ approach the SemWIQ [19] system contains contains a mediator service that distributes the execution of SPARQL queries. DAI-RDF [20] is a service-based RDF database middleware suite which extends the OGSA-DAI middleware to support RDF data processing activities including SPARQL query language, ontological primitives, and reasoning functions. Since DAI-RDF is based on service-based grid architecture, it is most promising to realize large-scale distributed scientific dataspaces.

Both approaches are feasible with the architecture. There might be use case scenarios and application domains where one approach fits better due to scale of the dataspace infrastructure or legal issues of participating institutions, etc. Also, a hybrid approach is plausible, for instance, when multiple already deployed dataspaces of homogenous application domains will be merged into a large scale dataspace infrastructure. We discuss such a scenario in Section 5.

3.3 Scientific Dataspace

While dataspace relationships are stored in the RDF store, the dataspace participants are organized in multiple heterogeneous databases that might be geographically distributed. In jSpace a participant represents a data set that either is the input data to a scientific experiment, or the analytical method being used within an experiment, or it is a dataset that has emerged during execution of an experiment. We therefore classify three types of participants: (a) primary data participants - the input data set, (b) background data participants i.e. an analytical method (web service, MATLAB script, etc.), and (c) derived data participants - emerged data sets i.e. histograms. The decision what DBMS to select for storing those different types of dataspace participants depends on the schemas of the corresponding data sets, to be used by the scientific community, to whom the dataspace is deployed. The OGSA-DAI [21] middleware is being used as common interface for all dataspace participants. Thus relational, XML, and file based resources can be organized as dataspace participants.

Dataspace participants are interconnected by relationships, which provide semantic information about the participant and the LCR they are connected to. Meta data of participants is organized by the OWL class *metaData* of the e-Science life cycle ontology. It allows the scientist to describe data sets according to user-defined attributes. An instance of the class *metaData* has the form of

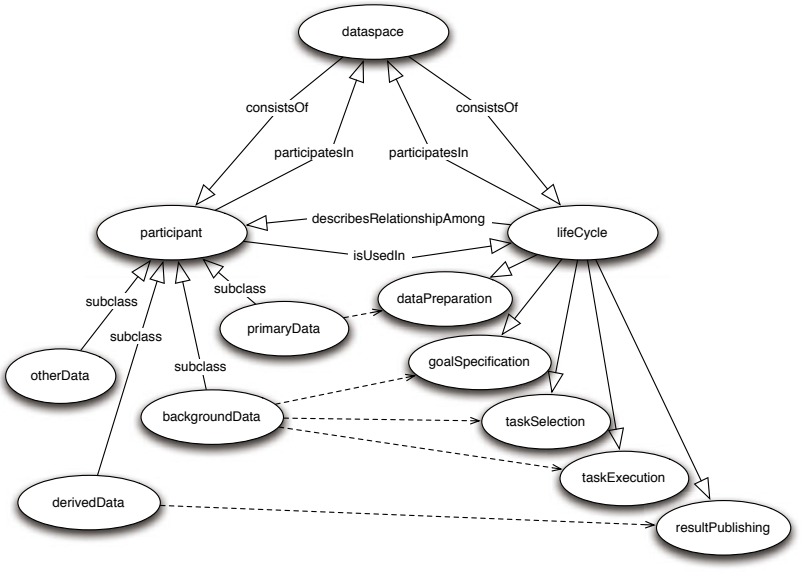

Fig. 2. Scientific dataspace modeled in the e-Science Life Cycle ontology

a triple `<instanceID, attribute, value>`. In Figure 2 we illustrate the main OWL classes and properties defined in the e-Science life cycle ontology showing how the scientific dataspace is modeled. It shows that the class *dataspace* consists of a class *participant* and a class named *lifeCycle*, which in fact represents relationships among participants. Instances of the lifeCycle class model how data sets (primary, background, or derived participants) were used in a specific experiment.

3.4 Dataspace Indexer

The purpose of the Dataspace Indexer is to organize LCRs, including their subscription. It implements a storage and indexing mechanism that allows to quickly evaluate the state of the dataspace in terms of calculating specific measures such as (total number of unique LCRs, number of reran LCRs, etc). Also questions like - *What activity was re-used most?* - can simply be answered without the need to access any RDF store. The examination of dataspace measures allows to monitor system usage and thus helps improving the system. The LCR indexes are organized in a flat table. Each row in the table represents a LCR key, which identifies the index of the resource itself and the indexes of all its participating individuals of the e-Science life cycle activities.

3.5 Search and Query Processor

Searching and querying a dataspace in general is not like querying a database. In a dataspace we need to drift away from the one-shot query to query-by-navigation.

Users will have to pose several queries, which results in an *Information Gathering Task* (IGT). IGT was introduced by Halevy et al. in [2] as one of the major principles of a dataspace system. In jSpace this task is implemented as a multi-level process where different types of queries can be submitted. In level 1 the RDF-Store, which organizes individuals of the e-Science life cycle ontology, is queried using SPARQL queries. The information a scientist is gathering in this first level represent semantics about applied scientific experiments, like what were the research goals, what data set was used, what analytical methods, etc. It will lead the scientist to those LCRs he might be interested in and to those that are interconnected to them. In the second level data items that are used within previously identified LCRs can be retrieved, by submitting queries to data sources that are participating the dataspace. Such data sets are for example the input data set used, or the data set derived from selected scientific experiments. In order to apply such kind of deeper searching and querying more sophisticated queries are submitted to the scientific dataspace, in particular to the corresponding DBMS that participates in the dataspace. Such level-2 queries can be in any other query language that is supported by the underlying data source.

Dataspace relationships and participants are precisely described by individuals of the e-Science life cycle ontology, therefore organized as RDF resources. The *Search&Query Processor* consists of a *Query Interpreter* and a *Query Translator*. The query interpreter receives a request, which can be expressed either as a SPARQL-Query or as keyword based search string or in any query language format that is supported by the underlying participants of the dataspace. For level-1 queries, the request is forwarded to the *Query Translator*, who generates a SPARQL query (if not yet already expressed in SPARQL) search string. This SPARQL query is then submitted to the RDF store. Level-2 queries are directly submitted to the OGSA-DAI client of the dataspace participant.

3.6 Dataspace Browser

The dataspace browser is a tool that allows the user to navigate trough the LCRs available in the dataspace in a visual way. It sends requests to the *Query Processor* in terms of SPARQL queries to be submitted to the RDF store. The response represents RDF data and is used as input for the dataspace browser.

There are a number of tools available that visualize RDF data. Some example projects include Welkin [22], multiple plugin tools for the Protege environment [23], and Semantic Analytics Visualization (SAV) [24]. These tools need to be elaborated and probably an appropriate tool might be adapted or some bits of the tools might be reused. However the jSpace architecture allows to easily attach own tools for browsing the dataspace. The decision what tools to use might depend on the community the dataspace system is deployed for.

4 Implementation Status

A first prototype of jSpace has beed applied to a small research group of a leading breath gas research institute [25], which is acting as an early core of adopters.

Guidelines defining mandatory descriptions for breath gas experiments were elaborated. This first prototype is based on the Jena framework with MySQL databases to provide persistent RDF data storage. We used the persistent ontology model provided in the Jena framework in order to create and store LCRs according to the concepts defined in the e-Science Life Cycle ontology. A dataspace client has a local copy of the ontology, which is used by the *RDF Store Connection Manager* to create a local ontology model. This model is then used to create new individuals and properties according to the ontology. Three MySQL databases for storing primary, background, and derived datasets were set up as OGSA-DAI resources, to be accessible on a service-based grid architecture.

The e-Science life cycle composer, a tool implemented in Java provides an easy graphical user interface to the breath gas researcher allowing to describe and publish breath gas experiments. It organizes text fields for pre-defined descriptions of breath gas experiments in five tabs according to their activity. For instance the *TaskSelection* activity, shown in the e-Science Life Cycle Composer GUI in Figure 3, requires to fill in a brief textual description and some corresponding keywords and to upload an archive file of the analytical methods being used in the experiment. The acting breath gas research group mainly uses MATLAB for their calculations. A typical background data set therefore is the collection of MATLAB functions used in an experiment compressed as zip archive. Once an experiment has beed finished, it can be published into the scientific dataspace by a single click on the OK button of the GUI. The information entered is used to create a new LCR, which then is saved in the corresponding RDF store. Connection details about local and global RDF Store as well as about corresponding OGSA-DAI resources are stored in a configuration file. A web service that communicates with an OGSA-DAI client stores the uploaded datasets into their corresponding databases.

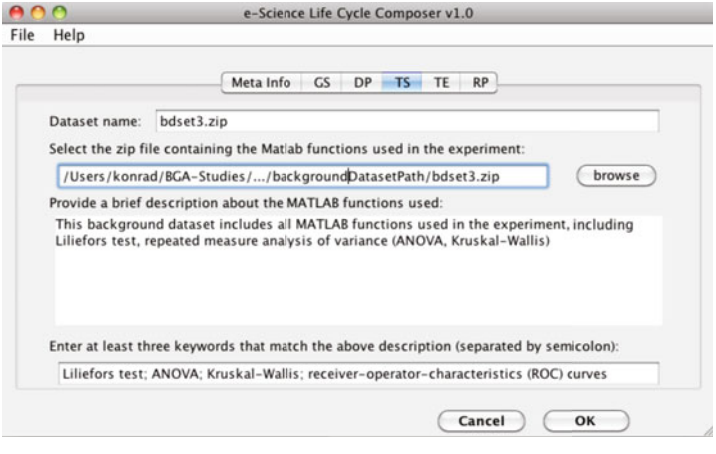

Fig. 3. The e-Science life cycle composer GUI

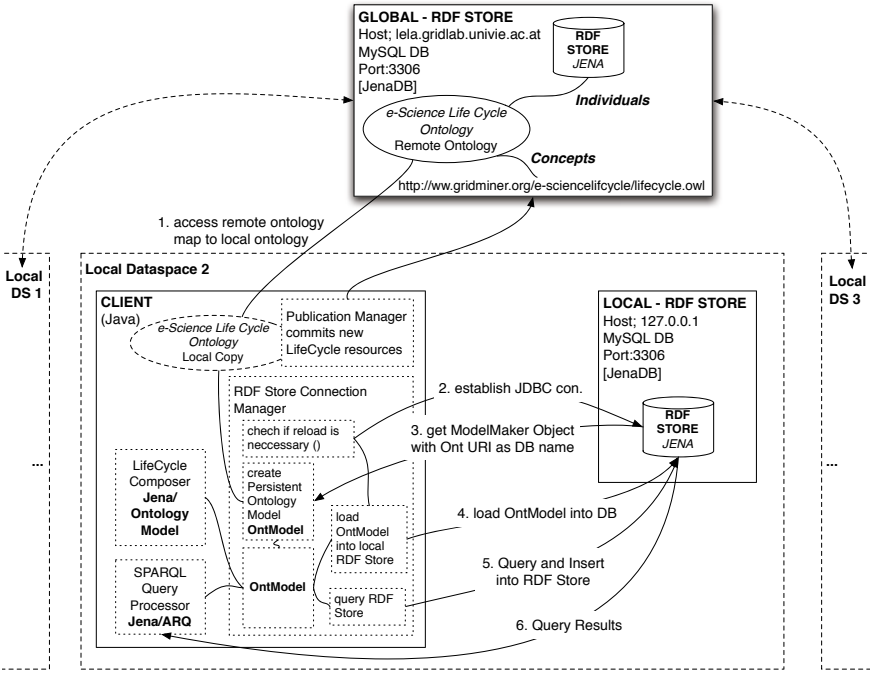

Fig. 4. Local and global RDF Store in a Scientific Dataspace Environment

In the first jSpace version we have chosen the data warehouse alike approach in order to provide efficient access to scientific experiments that were conducted at different research centers. However at the moment we only simulate a second dataspace instance. The *Connection Manager* handles a connection to a global and one to a local RDF store. Figure 4 illustrates one global RDF store, where meta data of breath gas experiments (LCR) that are public to the community is stored. There are three scientific dataspace instances illustrated, each deployed for a specific research group, which might be geographically distributed. Every dataspace instance has their own local RDF store to organize meta data of experiments that should be available only for the local organization. A *Publication/Update Manager* commits new LCRs to the global store. On the dataspace participants layer each dataspace instance deploys at least three databases as OGSA-DAI resources, where data sets used in an experiment are stored. Due to limit of space, this is not illustrated in the figure.

The e-Science life cycle model [9] has been applied as relationship model for the scientific dataspace. Jena SDB Version 1.3.1 with MySQL Version 5.0.67 as underlying RDBMS is used to implement multiple local and one global RDF stores. For the search and query interface we provided Joseki Version 3.4 as HTTP interface. A number of most important queries, such as {*Get me all experiments with VOC 'keyword'*}, and {*Get me all experiments from researcher 'name' where specified goal includes 'keyword'*}, or {*Get me all experiments with*

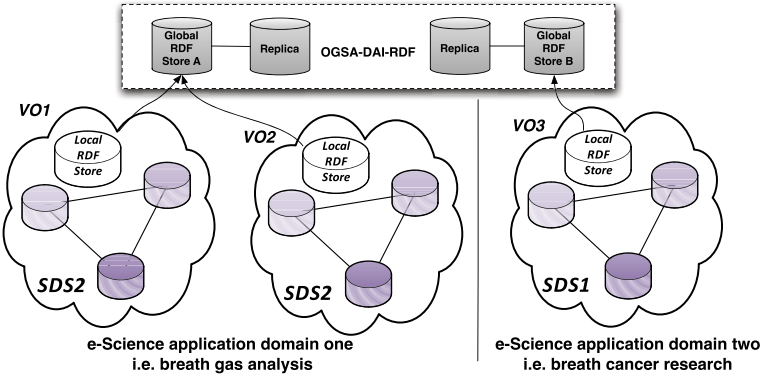

Fig. 5. Large-scale scientific dataspace infrastructure

ANY keyword equals 'keyword' and input dataset 'datasetName' is used} were predefined in SPARQL to enable the breath gas researcher to easy interact with the SPARQL query interface. However, the scientific dataspace is still in an early stage of evolution. It will need some time in order to get a large amount of LCRs including its corresponding dataspace participants into the dataspace.

5 Discussion

Currently, three breath gas research teams from our driving e-Science application produce at an average 48 breath gas experiment in three different studies in a single week. The total size of a LCR of a typical breath gas experiment including its primary, derived, and background data sets amounts to approximately 7 MB. The average number of triples that correspond to a single LCR is 170 with an average size of 150 KB stored in Jena's SDB triple layout. From this we can estimate the total size of the scientific dataspace after six month with a single research lab involved to roughly 8.7 GB with about 8160 triples in the local RDF store (approx. 50K triples in three years). In regard to the Berlin SPARQL benchmark [26], which states that the overall runtime for executing 50 query mixes on a 250K triples Jena SDB test data set is short above one minute, we can be confident that the presented solution provides reasonable performance. Based on this high-level estimation jSpace might need to scale up not before three years of deployment. However having multiple organizations or even various related e-Science application domains involved, it might be much earlier.

Vertical scalability can be achieved by interconnecting multiple dataspace instances, which leads to a large-scale scientific data space infrastructure. Such a scenario is illustrated in Figure 5. In this scenario we assume that each dataspace was already deployed for a specific virtual organization where acting scientist are feeding the dataspace continuously with their regularly running experiments. The global centralized RDF store approach was chosen to support exchange

with a second virtual organization of the same domain. Now, as both research domains are related scientific fields, it might be the case that scientists would like to share their knowledge among each other. In order to utilize this arising large-scale dataspace it will be necessary to provide a distributed RDF storage solution on top of global RDF stores. Most promising candidate to realize this is DAI-RDF, since it is based on OGSA-DAI, which we already use for organizing dataspace participants.

We are aware that we rely on active participation of members from the scientific community in order to establish a large scale scientific dataspace for breath gas analysis. Therefore we provide a simple interface that can easily be used by scientists from diverse research domains, especially for non-computer scientists, which was a major requirement from our driving application. However, we suspect that young-researchers (Master and PhD students) will be the major user group of the e-Science life cycle composer, while senior researcher will most likely interact with the system in terms of submitting requests. Once a first release is ready we expect that it enforces building of collaborations among breath gas research institutions as it supports the community in exchanging data and knowledge. This will build the basis for automation-based breath gas analysis.

6 Conclusions and Future Work

This paper presented the architecture of a scientific dataspace paradigm build on top of the e-Science life cycle ontology. A first prototype was developed on top of existing semantic web technology and deployed to a small core of early adopters in a highly relevant life science domain from which specific use cases are derived. It is the bases for development of an intelligent and more powerful second prototype, which will be based on semantic grid technology. One of the most promising technology for distributed SPARQL query processing in a large-scale context, which is a key concern for a wider acceptance of the presented dataspace paradigm seems to be the RDF(S) Realization [27], of the Data Access and Integration Working Group of the Open Grid Forum. However, as our prototype has shown that in order to get semantically rich scientific data from e-Science applications that can be further used we need first to ensure that the experiments being conducted are well preserved. Preservation of scientific data can best be established if the full life cycle of data is addressed [28]. This goal was successfully implemented by the first prototype of jSpace. We see in our future work both, to improve and automatize the preservation process and the upgrade from semantic web to semantic grid technology, which provides more powerful middleware for distributed management of storage systems that expose SPARQL endpoints.

References

1. Franklin, M., Halevy, A., Maier, D.: From databases to dataspaces: A new abstraction for information management. In: SIGMOD (2005)
2. Halevy, A., et al.: Principles of dataspace systems. In: PODS (2006)
3. Dong, X., Halevy, A.: Indexing dataspaces. In: SIGMOD, pp. 43–54 (2007)

4. Jeffery, S.R., Franklin, M.J., Halevy, A.Y.: Pay-as-you-go user feedback for dataspace systems. In: SIGMOD, pp. 847–860 (2008)
5. Das Sarma, A., Dong, X., Halevy, A.: Bootstrapping pay-as-you-go data integration systems. In: SIGMOD, pp. 861–874 (2008)
6. Dittrich, J.P., et al.: Imemex: escapes from the personal information jungle. In: VLDB. VLDB Endowment, pp. 1306–1309 (2005)
7. Li, Y., et al.: Research on personal dataspace management. In: IDAR, pp. 7–12 (2008)
8. Simmhan, Y.L., Plale, B., Gannon, D.: A survey of data provenance in e-science. SIGMOD Rec. 34(3), 31–36 (2005)
9. Elsayed, I., et al.: Intelligent Dataspaces for e-Science. In: CIMMACS, WSEAS, pp. 94–100 (2008)
10. Amann, A., et al.: Applications of breath gas analysis in medicine. International Journal of Mass Spectrometry 239, 227–233 (12 2004/12/15/print)
11. Elsayed, I., et al.: Towards realization of scientific dataspaces for the breath gas analysis research community. In: IWPLS, CEUR, UK (2009)
12. Elsayed, I., et al.: The e-science life cycle ontology (owl documentation) (2008), http://www.gridminer.org/e-sciencelifecycle/owldoc/
13. W3C: Resource description framework, RDF (2003), http://www.w3.org/RDF/
14. W3C: Web ontology language, OWL (2004), http://www.w3.org/2004/OWL/
15. Dittrich, J.P., Salles, M.A.V.: IDM: a unified and versatile data model for personal dataspace management. In: VLDB. VLDB Endowment, pp. 367–378 (2006)
16. Jin, L., Zhang, Y., Ye, X.: An extensible data model with security support for dataspace management. In: HPCC, pp. 556–563 (2008)
17. Prud'hommeaux, E., Seaborne, A.: SPARQL query language for RDF (2008), http://www.w3.org/TR/rdf-sparql-query/
18. Quilitz, B., Leser, U.: Querying distributed RDF data sources with SPARQL. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 524–538. Springer, Heidelberg (2008)
19. Langegger, A., et al.: A semantic web middleware for virtual data integration on the web. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 493–507. Springer, Heidelberg (2008)
20. Kojima, I., et al.: Implementation of a service-based grid middleware for accessing RDF databases. In: Meersman, R., Herrero, P., Dillon, T. (eds.) OTM 2009 Workshops. LNCS, vol. 5872, pp. 866–876. Springer, Heidelberg (2009)
21. Antonioletti, M., et al.: OGSA-DAI 3.0 - the whats and the whys. In: Proceedings of the UK e-Science All Hands Meeting 2007 (September 2007)
22. Mazzocchi, S., et al.: Welkin - a graph-based RDF visualizer (2004), http://simile.mit.edu/welkin/
23. Protege: a free, open source ontology editor and knowledge-base framework (2010), http://protege.stanford.edu/
24. Deligiannidis, L., et al.: Semantic analytics visualization. In: Mehrotra, S., Zeng, D.D., Chen, H., Thuraisingham, B., Wang, F.-Y. (eds.) ISI 2006. LNCS, vol. 3975, pp. 48–59. Springer, Heidelberg (2006)
25. Amann, A., et al.: Volatile organic compounds research group (2009), http://www.voc-research.at/
26. Bizer, C., et al.: The berlin sparql benchmark. Int. J. Semantic Web Inf. Syst. 5(2), 1–24 (2009)
27. Gutiérrez, E., et al.: Accessing RDF(S) data resources in service-based grid infrastructures. Concurr. Comput.: Pract. Exper. 21(8), 1029–1051 (2009)
28. Lynch, C.: Big data: How do your data grow? Nature 455(7209), 28–29 (2008)