2012 Eighth International Conference on Semantics, Knowledge and Grids

The VPH-Share Data Management Platform: Enabling Collaborative Data Management for the Virtual Physiological Human Community

Martin Koehler [#1], Richard Knight [*2], Siegfried Benkner [#3], Yuriy Kaniovskyi [#4], Steven Wood [*5]

[#] *University of Vienna, Research Group Scientific Computing*
1090 Vienna, Austria
[1] koehler@par.univie.ac.at
[3] sigi@par.univie.ac.at
[4] yk@par.univie.ac.at

[*] *Scientific Computing & Informatics, Sheffield Teaching Hospitals NHS Foundation Trust*
Sheffield, UK
[2] richard.knight@sth.nhs.uk
[5] steven.wood@sth.nhs.uk

***Abstract*—The objective of the Virtual Physiological Human Initiative is to provide a systematic framework for understanding physiological processes in the human body in terms of anatomical structure and biophysical mechanisms across multiple length and time scales. The VPH-Share project, which has been funded in the context of the initiative, contributes to this vision, especially in terms of data management. In this paper we present the VPH-Share data management platform enabling sharing VPH-relevant datasets within the community on the basis of Cloud technologies. The data management platform aims at supporting the data management life-cycle but starting from already available data. The life cycle covers all processes from data selection, semantic data annotation, data integration, data publishing, and data access. Herein we describe the infrastructure supporting the data management life-cycle towards collaborative data management, consisting of the data publication suite and the dataset service environment.**

I. Introduction

The Virtual Physiological Human Initiative (VPH-I) from the European Commission focuses on the provisioning of a systematic framework for understanding physiological processes in the human body in terms of anatomical structure and biophysical mechanisms at multiple length and time scales. The European Project VPH-Share, funded within the VPH initiative, aims to contribute to this objective by transforming the European health care system into a more personalized, predictive, and integrative process with significant impact on health care and on disease prevention. Hence, the project develops an integrated framework enabling the exposure and management of data, information, and tools leading to the composition of new VPH-related workflows and to profound collaborations between its members. As starting point, the project addresses four flagship workflows resulting from the European projects @neurIST [1], Virolab [2], euHeart [3], and VPHOP [4] which provide existing data, tools, and models driving the development of the infostructure.

The vast amount and increasing complexity of biomedical information that has been consented for research by the VPH initiative outruns the current practice of efficiently managing and sharing this information. To address this challenge, the VPH-Share project develops a unified data management platform (DMP) [5] comprising generic services and protocols to enable data controllers to manage the provisioning and sharing of the information. The DMP follows a process similar to incremental Extract-Transform-Load (ETL) [6] resulting in an evolving data set as additional data sources become available at time. Additionally, the platform enables the creation of on-demand customized dataspaces utilizing pay-as-you-go semantic data integration.

Additionally, the VPH-Share project develops a Cloud-based infrastructure enabling the execution of workflows and applications as well as the provisioning and sharing of data sets within the community. The DMP exposes its data sources on top of this Cloud-based environment by utilizing the concept of semantic and atomic services [7]. Atomic dataset services enable the management of a dataset and are built upon virtual appliances and a generic Web service environment. The focus of the DMP is on (semantically annotated) relational data although data is often available in a simpler form of data structure such as Microsoft Excel or CSV files.

The DMP enables on-demand management and provisioning of (new) data sources, which includes adding and removing data sources, semantic annotation of data sources, as well as access to and integration of them. The platform includes the VPH-Share dataset service environment for on-demand exposure of datasets as Cloud-services. The VPH-Share dataset service environment provides SQL-based interfaces enabling relational data access and support distributed data mediation approaches following the Global-as-View (GaV) approach [8]. Additionally, semantic data access is supported on top of data annotations, SPARQL, and the Linked Data approach [9].

978-0-7695-4794-7/12 $26.00 © 2012 IEEE
DOI 10.1109/SKG.2012.51

Given this context the objective of the DMP release is to provide uniform service interfaces and protocols:

- Expose data sources as Cloud-services
- Enable relational querying of data sources
- Enable semantic access to data sources
- Provisioning of data services in the Cloud
- Automatic management of dataset in the Cloud
- Provide global views across multiple data sources (semantic data mediation)

In the following chapters a detailed description of the DMP, the comprised components, and the utilized methodologies is given. Chapter II explains the DMP as a whole. Chapter III goes into detail about the data publication mechanisms and Chapter IV delineates the dataset service environment. Afterwards, the semantic data integration approach on the basis of the Linked Data concept is described. Finally, related work, and concluding remarks including future plans are presented.

II. Data Management Platform Architecture

The DMP subsumes the capabilities towards data management, access, and integrating within the VPH-Share project. The DMP consists of the data publication suite, the dataset service environment, and the linked data environment. The data publication suite enables the ontological annotation of data sets, the de-identification of the data, and the publishing of the data set in the VPH-Share Cloud environment. The dataset service environment is utilized for Cloud-enabling datasets and for exposing them via RESTful and Web service interfaces. Additionally, it supports the provisioning of virtual data sets mediating multiple possibly heterogeneous data sets each exposed as dataset service as well. The Linked Data environment is utilized in addition to enable traversal of exposed data sets on top of ontological concepts.

Therefore, the fulfillment of the described objectives within the DMP has been achieved on top numerous technologies. All interactions between users or tools and the DMP follow standard protocols accessible via the Web. In multiple cases, different endpoints providing Web service or RESTful interfaces are supported.

Besides the technical issues how data can be managed and accessed within the community, a lot of legal and ethical constraints have to be met. Normally, data to be shared within the community is created and available within one organization and is most often sourced from a clinical system. This implies that data is often related to patients and therefore is fully identifiable. For this reason, these data sets can not be easily made available to the community without issuing the following challenges:

1) Data is hosted in hospital local networks (no Internet access)
2) Data protection and ethics laws prohibit data sharing
3) Securing the data is not possible
4) Data access cannot be audited
5) Data has no meaning for external users

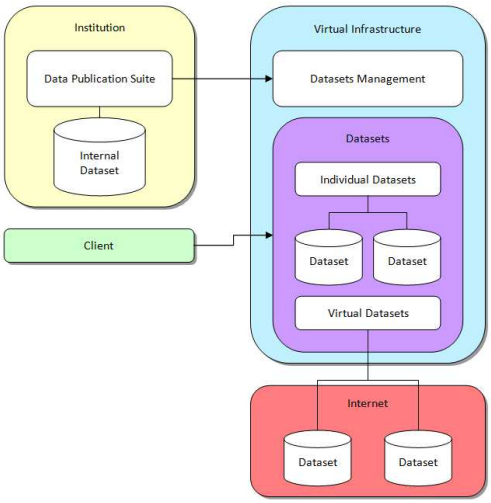

Fig. 1. Data Management Platform: Individual and Virtual Datasets

The DMP aims at providing a technical platform bypassing these drawbacks. The first step towards collaborative data sharing in the community is therefore to give the data a meaning by annotating the fields of the data schema with ontological terms. In this step the data fields have to be de-identified as well. This process is assisted by the data publication suite (DPS). The DPS has to be installed inside the institution to enable access to the given data source. Additionally, the DPS follows a modular architecture supporting the integration of different types data sources. The only limitation is that it must be programmable by developers available to the institution. As starting point, the DPS supports a core set of data sources and de-identification tools. Additional functionality can be included by creating extra modules.

After the data annotation and de-identification process, the DPS enables publishing of the data sets to a secure place within the VPH-Share Cloud which will be accessible via the Internet. By utilizing the VPH-Share security architecture, this approach will enable authorized access to data sets by different stakeholders. Data published in this fashion is referred to as an "Individual Dataset", since there is one dataset stored in the DMP for each dataset published through the DPS. Additionally, the DPS will support the provisioning of "Virtual Datasets" enabling linking to external datasets which are outside the scope of the DMP. The access mechanisms for virtual datasets will be the same as for individual datasets. Supplementary, a virtual dataset enables the provisioning of a global views integrating multiple individual (or other virtual) datasets. This approach supports the provisioning of tailor-made views on top of multiple datasets for specific stakeholders. Of course, virtual datasets will include the same security guidelines and enable only authorized access. An overview of individual and virtual datasets is depicted in Fig. 1. The institution itself hosts an internal dataset which can be exposed to the VPH-Share community via the DPS. These exposed datasets are referred to as individual datasets and are managed

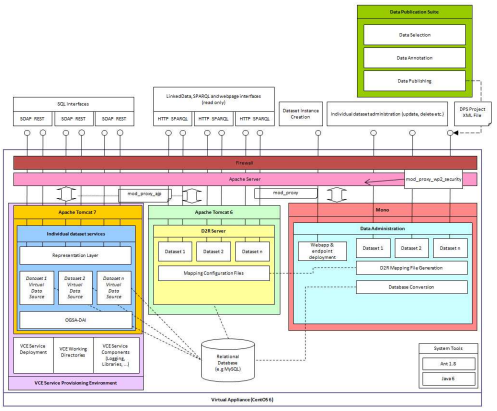

Fig. 2. Data Management Platform: Architecture of Virtual Appliance

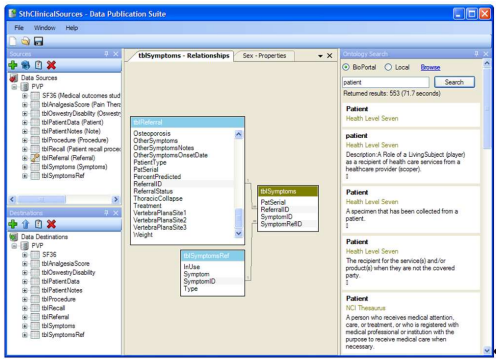

Fig. 3. Data Publication Suite: Shows data sources, data destination for publishing the source, the schema and relationships, and ontological search

by the DMP. All datasets (individual and virtual) are made available via the VPH-Share dataset service environment.

The architecture of the DMP is depicted in Fig. 2 and is provided as virtual appliance. By following the concept of virtual appliances, this architecture enables on-demand deployment of new datasets within the VPH initiative and hosting of them in the VPH-Share Cloud. Additionally, this approach adds flexibility in the hardware requirements (e.g. disk space) due to the capabilities of the Cloud. The DMP virtual appliance comprises installations of the dataset service environment, the Linked Data environment, and the data administration tools as well as an installation of a relational database. The dataset service environment exposes Web and RESTful service interfaces for the exposed individual datasets while the D2R server [10] is utilized for a linked data view on the datasets. The data administration tools enable the management of the individual datasets to be exposed and the annotation of the data residing in the relational database.

This architecture enables different views on the data and multiple query mechanisms are supported. By utilizing the SQL endpoints data can be easily exported and utilized in traditional tools. The SPARQL endpoints can be used for accessing data by means of established ontological concepts. Additionally, the data is exposed via the Linked Data concept.

III. Data Publication Suite

The Data Publication Suite (DPS) resides within the VOH-Share related institution and allows VPH-Share data providers to ontologically annotate numerous datasets, de-identify the data and publish them to a centrally accessible location on the VPH-Share cloud. The exposed datasets will not be open to any user of the Internet but only to users who have been authenticated by VPH-Share and who have agreed to the terms and conditions, set by the data provider and attached to the use of the data. The design of the DPS has been orientated around the use of graphical interactions, as opposed to text configuration. This enables a larger outreach into the community of people able to publish their data and around modules so the system can be extended to fit the needs of all users. Specifically, the modular design enables connections to custom data sources and the use of custom de-identification algorithms. The DPS is the only client software of the DMP which is needed for publishing fully semantically annotated and de-identified datasets to a secure location on the internet. Summarized the DPS enables selecting data from the internal data source, management of data relationships, the annotation of the data, and publishing of datasets in the VPH-Share Cloud.

A. Data Selection

The data selection process starts with choosing a a “data source” which defines any storage system from which data can be extracted. The place where datasets are hosted, VPH Share storage instances, are defines as “Destinations”. Within the DPS, each type of data source is individually developed as a module which enables individual institutions to develop their own. This allows them to connect with their own custom systems and publish data from systems which aren’t supported by the actual version of the DPS. Currently, modules have already been created for Microsoft SQL Databases and files and databases which support the OLE DB API (i.e. CSV files, Excel spread sheets, Access databases).

When a new data source is added to the DPS, the connection information has to be entered. Afterwards, the schema of the dataset is extracted and displayed in a hierarchy below the source node. Users are enabled to select parts from the database schema which should be exposed as an individual dataset.

B. Data Relationships

When dealing with relational databases or multiple single table sources, there needs to be relationships between the data so it can be presented to the user in a comprehendible fashion. In most cases we have encountered the database doesn’t store any relationships and it is simply acts as an unrelated table storage area while the relationships and constraints are built into the application presented to the user. Where these relationships exist they will be extracted from the source and into the DPS when the source is added or updated but for

instances where they don't we needed to produce a graphical way to create, view and edit relationships.

C. Data Annotation

After the dataset to be exposed has been selected and the relationships have been added, the dataset can be annotated with ontological concepts from multiple well established domain-specific ontologies. The annotation process starts with searching for existing ontological. Users are enabled to specify free text and the DPS suggests concepts from a predefined set of ontologies. The free text would normally be similar to that of the column name of the database, that is, if the columns have been sensibly named. The ontological search utilized the BioPortal REST API which allows an extensive search of 300 ontologies [11]. In the moment the set of ontologies has been limited to the following ontologies to reduce the number of terms returned: SNOMED Clinical Terms, NCI Thesaurus, NCI Metathesaurus, HL7, ICD-9.

This collection has covered our needs for development purposes but should not be considered as a list of the best ontologies to use. Future work will include the implementation of a project specific ontology repository and a better search facility than BioPortal whose limitations are search speed and result ranking. The search ranking used by BioPortal is purely a best text match system. VPH-Share requires a more complicated ranking algorithm which will increase the rank of terms which are better mapped to other ontologies giving better results when users are searching for data.

The ontology search window displays the preferred name, ontology and description of each term returned by the search. Once these results have been populated, the terms can be dragged from here to multiple places in the DPS, mainly a field or table in the sources window. Ideally each item should have only a single annotation but in some cases the ontology repository doesn't contain a single term to represent the item completely but normally provides sufficient terms, when used in combination, to describe it. Combining these terms correctly in an ontological language to give the intended description is difficult and is normally only attempted by ontology engineers. This process will be covered in future work in the VPH-Share project.

Once all the tables and fields have been annotated, it is now necessary to annotate some of the data, more specifically data which forms part of an enumerated list. For example, the field "Sex" has already been given the annotation "Gender" from the NCI Thesaurus ontology. This data included in this field could be further annotated since it contains a known set of values. In this case the gender of the patient has been stored as a string and by analyzing the data, have found only two enumerations, "M" and "F" which stand for male and female. In other databases the enumeration "F" could be represented in different ways such as "Female" or "2" and even though a human could understand the first and second and make an educated guess at the third, a computer doesn't have sufficient knowledge to infer the correct meaning. This limits the distributed search capabilities to a free text search reducing the number of results because the computer doesn't understand the search criteria and the best it can do is match the text. To give a more accurate set of search results the data must also be annotated with ontological terms. The DPS supports data annotation in the same way as fields, by dragging a term from the ontology search window onto the data enumeration in the fields' property window.

D. Data Publication

After datasets are selected, relationships have been added, and the data has been annotated, the source has been completely configured and can be published in the VPH-Share Cloud. Therefore, a destination, preconfigured sites in the VPH Cloud, has to be selected and a new virtual appliance is configured with the dataset and the appropriate dataset services. During the publication process the de-identification algorithms are utilized. Currently, the only de-identification implemented is the removal of fields, and simple SHA1 hashing of strings but in future releases more sophisticated algorithms will be implemented.

Additionally, the data publication process will also support the configuration of virtual datasets integrating multiple individual datasets depicting the importance of inter-source relationships. The publication process is based on data management services.

E. Dataset Management Services

The DPS provides dataset management services enabling the interaction with the server and supporting create and update of new dataset instances. Each virtual appliance provides one dataset management service endpoint to create a new instance, while update endpoints are provided for each exposed individual dataset. The dataset management services enable the creation of new datasets in the virtual appliance and automates the publishing process via the dataset service environment.

IV. Dataset and Virtual Distributed Dataset Services

The DMP distinguishes between services which are related to dataset access (individual datasets) and services provided for dataset integration (virtual datasets). Therefore, a middleware is provided in order to setup and deploy appropriate dataset services as well as virtual dataset services that offer transparent access to multiple, potentially heterogeneous data sources via a single uniform service interface.

Dataset services expose their capabilities via both RESTful and standard web service interfaces (SOAP, WSDL). Offering two well-established types of service interface technologies will simplify the integration of dataset services into client toolkits that are based on different programming languages. Dataset and virtual dataset services have been developed on the basis of the Vienna Cloud Environment (VCE) middleware [12], [13], [14], [15] and utilize technology developed in the context of OGSA-DAI [16] and are available via the Dataset Service Environment (DSE).

A. Dataset Services

Dataset services are utilized for providing remote access to a single data source via an OGSA-DAI compliant interface, supporting both read- and write-access. The implementation of dataset services is based both on standard Web services and RESTful technologies. Dataset services expose a REST-based interface enabling the execution of queries via plain HTTP methods. Additionally, a web service interface on the basis of the Apache CXF framework is provided. Therefore, a dataset service exposes two representation layers and requests can be sent to either of them.

Dataset services can be configured with support for relational data access via SQL as well as with semantic data access via SPARQL depending on the resource configuration. Currently, both endpoints are utilized to enable the utilization within different tool chains. Virtual dataset services support only SQL. The development of virtual datasets on the basis of semantic technologies will be tackled in future releases.

The query execution mechanism is based on the concept of tasks. A task represents the execution of a single query. This concept enables monitoring of executed queries on a task basis. Internally the dataset service uses OGSA-DAI [16] for accessing the data resource.

Two types of dataset services can be configured. Parameterized services enable the execution of prepared statements against the exposed data source. Generic dataset services enable the execution of OGSA-DAI perform documents against the virtualized data source.

1) Architecture of Dataset Services: A generic view of the dataset service architecture is depicted in Fig. 4. The VPH-Share service provisioning environment is installed within the virtual appliance and includes a service hosting environment (Apache Tomcat 7). The service provisioning environment consists of service deployment tools, working directories (managing the execution of queries), as well as additional configurable service components (logging, etc.).

Dataset services can configured in different ways. Normally, one Web and RESTful endpoint is exposed for each hosted dataset. Additionally, the service environment supports exposing multiple datasets via a single endpoint which is basically important in the context of virtual dataset services.

The RESTful interface provides generic functionality enabling the execution of tasks by means of the following methods:

- Method for uploading perform documents/parameter files (based on service configuration)
- Method for starting access to the data source (query execution)
- Method for querying the state of data access
- Method for downloading the results

Additionally, a WSDL interface is provided on the basis of OGSA-DAI services. Via the WSDL interface, a client can directly access the OGSA-DAI services and the OGSA-DAI data sources. An OGSA-DAI perform document can be uploaded and executed against a given data source.

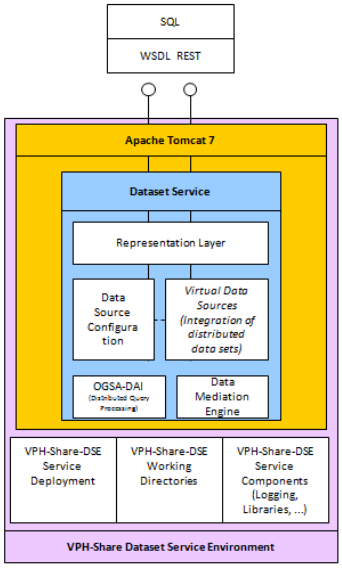

Fig. 4. Dataset Service Architecture

Dataset services are compliant with OGSA-DAI, the de-facto standard for data access and integration on the basis of web services, supporting access to different data sources including relational databases, flat files and XML databases. To send a query-based request to a dataset service the client uploads an OGSA-DAI perform document containing the SQL query. The OGSA-DAI perform document, an XML document, can be constructed on the basis of the integrated workflow builder tool provided with the VPH-Share client toolkit. It supports basic OGSA-DAI workflows for executing SQL queries and transforming the results into a WebRowSet format or into a CSV file. Furthermore, the OGSA-DAI client toolkit can be utilized for the creation of more complex workflows.

Parameterized dataset services support as input a SQL query or a set of ontological terms, depending on the service configuration. The result format can be configured as well. Parameterized dataset services support as result format the XML WebRowSet and CSV.

B. Virtual Dataset Services

A virtual dataset service is capable of exposing a single virtual view on multiple, heterogeneous data sources. Two or more heterogeneous individual (or virtual) datasets can be provided as a single virtual dataset on the basis of flexible data integration/mediation mechanisms. Currently, virtual datasets are based on expressing views via the SQL language. In future releases, data mediation will be enhanced by providing an XML schema mapping file which specifies how to decompose queries against a virtual schema into queries for the underlying target data sources. Virtual dataset services can also be used to provide different views on top of a single data source (e.g. presenting different parts of the data to varying stakeholders).

From the client's point of view data mediation is fully transparent, and as a consequence, the same interface and

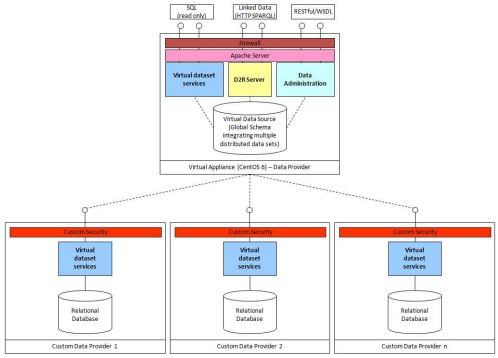

Fig. 5. Virtual Dataset Service Architecture

access mechanisms as for basic dataset services may be used. However, the current version of virtual dataset services support only read access.

1) Architecture of Virtual Dataset Services: A structural overview of the architecture of virtual dataset services is depicted in Fig. 5. Similar to common dataset services, the representation layer of virtual dataset services exposes a RESTful and a WSDL-based service interface. Virtual datasets are implemented on the basis of OGSA-DAI views and on top of the OGSA-DAI distributed query processing engine (DQP) [17]. The DQP engine is responsible for processing and distributing queries against individual dataset services it is connected to. OGSA-DAI views provide a global virtual table, as they decompose and filter the request query into a set of target queries against the integrated data sources. In case of a virtual dataset service these queries are executed by the DQP engine. Moreover, the DQP engine is responsible for combining the individual results from the target data sources into a single result.

A virtual dataset service requires a view definition file or a mediation schema (GDMS) [18], [13] for establishing a virtual (mediated) data source. The virtual view defines a relation, specifying how the individual data sources are mapped to the mediated global view by following the Global as View approach. Providing a global view means tight federation, which offers schema, language and interface transparency. The virtual dataset service follows the virtual integration approach, ensuring to query always up-to-date data.

The current implementation of virtual dataset services requires static binding of data sources. Current developments are aimed at providing a more dynamic, semantics-enhanced model to support on-the-fly data integration and transformation.

C. Deployment of Dataset and Virtual Dataset Services

The dataset service provisioning environment includes a graphical and command line deployment tools supporting the automatic deployment of dataset services and virtual dataset services.

The graphical deployment tool supports the configuration of dataset services and virtual dataset services and the exposed data sources in a graphical manner. Users are enabled to select one or multiple datasets to be exposed, configure the type of dataset service, the utilization of data mediation, and additional service capabilities such as security or an additional administrative Web portal.

Additionally, the dataset service environment provides an automatic deployment tool which is utilized by the DPS for publishing new individual datasets without user interaction. This tool enables to automate the process of deploying and undeploying dataset services on demand.

D. Service Interaction Scenarios

The VPH-Share DMP release includes a client toolkit enabling access to dataset services and virtual dataset services. The client toolkit consists of a high level Client API enabling the execution of queries on the basis of SQL and SPARQL, a command line client, and a Web-based dataset browser. The Client API can be easily utilized within applications and is provided in multiple programming languages (Java, C#, Python) The command line client enables access to dataset services and virtual dataset services via the command line which can be used for automated scripts. Finally, a dataset browser enabling the presentation of the schema of the dataset, the selection of parts of the schema resulting in the presentation of the data as well as the execution of SPARQL and SQL queries is provided. In the following, the different client mechanisms are described in more detail.

1) Description of Client API: The client API provides a basic implementation for accessing dataset services. The user has to define the query to be executed against the service. Afterwards, the query execution can be started by utilizing the executeQuery() method. The actual state of the query execution can be retrieved via the getState() method. During query execution the state will be set to PROCESSING. After the query has finished the state will be set to FINISHED. Multiple implementations of the getResults() method are available for retrieving the results of a query. The default implementation writes the results into an output file. The format of the data is specified by the utilized workflow or in case of a parameterized service, by the service configuration.

2) REST-based Service Interaction: Additionally to the provided Client toolkit, the REST-based interface can be called via a HTTP client. HTTP client implementations are available for almost any programming language enabling easy integration of dataset service invocations into all types of applications.

As previously described, the REST interface has been implemented on the basis of tasks representing the execution of one query against one specified data resource. The basic steps of executing a query include uploading an input file, starting the query execution, querying the state of the execution, and downloading the results. The main RESTful resource represents the tasks managed by the service. Each task is represented as a resource itself. The task representation includes the actual state of the task and all input and output

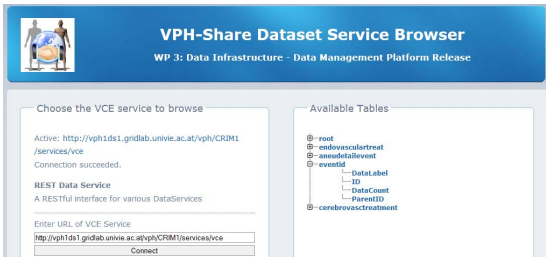

Fig. 6. Databrowser: Web-based access to individual and virtual dataset service

files aligned to this task. Input and output files are presented as resources as well.

By querying a RESTful VCE service, the user is able to select the representation type of the results. The current version supports two representation types, including XML and HTML. The XML representation is utilized by the VCE client implementation. The HTML representation enables the utilization of a VCE service via a web browser.

The resources are accessed through the well-known HTTP methods.

3) Browser-based Access: The dataset browser enables browsing of available datasets, the schemas, as well as the result by users (see Fig. 6). In future version users shall be enabled to search for specific services by utilizing ontological terms. The current version enables the selection of available datasets. This results in the presentation of the relational schema in the Web browser. The user is enabled to select columns of a specific table to be queried which results in the presentation of the data.

Additionally, the data browser enables querying of dataset services via SPARQL. In this case, the browser depicts the ontological concepts utilized for data annotation of this dataset. The user can select a concept and the data will be presented.

Besides the graphical query mechanism, the users are enabled to insert their own SQL or SPARQL queries to be executed against the dataset service.

V. Linked Data Services

SPARQL, standing for SPARQL Protocol and RDF Query Language, enables the querying of Resource Description Framework (RDF) data sources. Linked Data enables the retrieval of RDF information about a class via a URI, it is used to uniquely identify ontological terms but also allows the terms' details to be downloaded in RDF format. In our case it will be used to identify specific data rows, such as a single patient. Both of these concepts are core to the semantic Internet [9].

To achieve semantic interactions the open source project called D2R Server [10] has been utilized. This bridges the gap between relational database and the semantic world using a mapping to translate between SQL databases and RDF triples. The D2R Server offers extra built-in functionality: the ability to browse the RDF data through a web page rendered for human viewing (i.e. not RDF/XML) and an AJAX SPARQL client to run queries; all without any extra software which is highly useful for development.

The D2R Server enables different access methods and views on the data including a HTML view for human interaction, an RDF view conforming to the Linked Data specifications, and a SPARQL endpoint. As described, data can be accessed through the D2R server in Linked Data format. Linked Data must be accessible via a URI which returns a set of RDF triples defining the data; a semantic web browser should be used to interpret and display this data. Not only does this data contain the annotated published data but will also hold links to more data such as datasets pertaining to the same patient or extra data in the same dataset. Utilizing this approach enables human users to easily browse individual datasets. Additionally, the SPARQL endpoints will be employed by services and experienced users to extract specific data from across the hosted datasets of VPH-Share.

VI. Related Work

Multiple projects in the biomedical domain addressed the problem of providing an infrastructure for collaborative data management. The @neurIST project ealt with supporting the research and treatment of cerebral aneurysms and provided data access and data mediation services [15] on a relational basis and built a clinical reference information model (CRIM) enabling the provisioning of a a generic view on top of distributed and heterogeneous data sources. Nevertheless, the VPH-Share DMP tries to achieve a solution for the whole data management life-cycle, including the selection, the annotation, the publishing, and the access and integration of data in a generic manner.

Another project in the domain of biomedical data management, called DebugIT [19], builds a framework for integrating biomedical data from several clinical information systems (CIS) spread over Europe. They focus on the challenging tasks of managing heterogeneous data sources, privacy and security and utilize semantic technologies within their framework. While the data integration process follows a related apporach as the VPH-Share project, the VPH-Share project focuses on the whole life-cycle of data management and on the provisioning of a generic architecture applicable to different domains.

Additionally, the project Health-e-Child [20] created an information modelling methodology based around three complementary concepts: data, metadata, and semantics. The objective of the project is to give clinicians a comprehensive view of a child's health by integrating biomedical data, information and knowledge. The utilized data spans from imaging to genetic to clinical and epidemiological.

The caCORE infrastructure [21] which has been developed by the National Cancer Institute (NCI), United States provides tools for the development of interoperable information management systems for data sharing and is particularly focused on biological data in the cancer domain and has been reused in additional projects.

VII. Conclusion and Future Plans

Currently, the VPH-Share project has implemented a DMP release including initial versions of the data publication suite, dataset services, and Linked Data. The DMP release enables the provisioning of distributed and heterogeneous data sources in the VPH-Share project on a service basis. The DPS supports the selection of datasets, the semantic annotation of those on the basis of ontologies, and their provisioning as services and linked data.

The actual version of the release represents the foundation for achieving the objectives such as exposing data sources in the Cloud, enabling semantic access to data, and automatic management of datasets. The architecture is based on virtual appliances and the deployment process has been developed according to the VPH-Share Cloud infrastructure. This approach enables to achieve a tight integration of the project internal tool chain.

A focus of future work will be on the integration of the security framework which is currently being developed according to the legal and ethical constraints given by the community. The current release does not consider security which might present an obstacle to its full utilization within the project.

Dataset services and Linked Data represent the foundation for distributed semantic data access and integration within the VPH-Share project. Currently, the project is working on the design and implementation of a distributed query processing (DQP) service which will support the distributed execution of semantic queries against distributed and semantically annotated datasets. This service should negate the need for users and services to directly interact with the dataset services and Linked Data services of the DMP, even though possible, when searching and retrieving data due to the added functionality while still enabling single dataset access. Linked Data will still be available as currently implemented, allowing the browsing of individual datasets through a semantic web browser.

Another major future issue is the interaction between the workflows developed within the European projects euHeart, @neurIST, VPHOP, and Virolab. The data browser interface as well as the query tools have to be adapted due the requirements arising from the specific workflow scenarios.

Acknowledgment

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2011-2015) under grant agreement #269978 (VPH-Share Project).

References

[1] H. Rajasekaran, P. Hasselmeyer, L. L. Iacono, J. Fingberg, P. Summers, S. Benkner, G. Engelbrecht, A. Arbona, A. Chiarini, C. Friedrich, M. Hofmann-Apitius, B. Moore, P. Bijlenga, J. Iavindrasana, H. Müller, R. Hose, R. Dunlop, A. Frangi, and K. Kumpf, "@neurIST - Towards a System Architecture for Advanced Disease Managment through Integration of Heterogeneous Data, Computing, and Complex Processing Services," in *IEEE International Symposium on Computer-Based Medical Systems*. Jyväskylä, Finland: IEEE Computer Society Press, June 2008, copyright (C) IEEE Computer Society.

[2] VIROLAB, "EU IST STREP Project, 027446, http://www.virolab.org/," 8 2011.

[3] EuHeart, "Integrated cardiac care using patient-specific cardiovascular modeling, http://www.euheart.eu," 8 2011.

[4] VPHOP, "The Osteoporotic Virtual Physiological Human: http://www.vphop.eu," 8 2011.

[5] S. Benkner, J. Bisbal, G. Engelbrecht, R. D. Hose, Y. Kaniovskyi, M. Köhler, C. Pedrinaci, and S. Wood, "Towards collaborative data management in the vph-share project," in *Proceedings of the Intl. Workshop on Cloud Computing Projects and Initiatives, in conjunction with Euro-Par 2011*. Bordeaux, France: Springer, Aug 2011.

[6] M. Franklin, A. Halevy, and D. Maier, "From databases to dataspaces: a new abstraction for information management," *SIGMOD Rec.*, vol. 34, no. 4, pp. 27–33, Dec. 2005.

[7] C. Pedrinaci and J. Domingue, "Toward the next wave of services: Linked services for the web of data," *Journal of Universal Computer Science*, vol. 16, no. 3, pp. 1694–1719, 2010.

[8] J. Ullman, "Information integration using logical views," in *Database Theory ICDT '97*, ser. Lecture Notes in Computer Science, F. Afrati and P. Kolaitis, Eds. Springer Berlin / Heidelberg, 1997, vol. 1186, pp. 19–40.

[9] C. Bizer, T. Heath, and T. Berners-Lee, "Linked Data - The Story So Far," *International Journal on Semantic Web and Information Systems (IJSWIS)*, 2009.

[10] Chris Bizer, "D2R Server: http://d2rq.org," 5 2012.

[11] BioPortal, "NCBO BioPortal, http://bioportal.bioontology.org," 5 2012.

[12] M. Köhler and S. Benkner, "VCE - A Versatile Cloud Environment for Scientific Applications," in *The Seventh International Conference on Autonomic and Autonomous Systems (ICAS 2011)*, Venice/Mestre, Italy, May 2011.

[13] M. Köhler and S. Benkner, "A service oriented approach for distributed data mediation on the grid," in *Grid and Cooperative Computing, 2009. GCC '09. Eighth International Conference on*, Lanzhou, Gansu, China, Aug 2009, pp. 401–408.

[14] S. Benkner, G. Engelbrecht, M. Köhler, and A. Wöhrer, "Virtualizing Scientific Applications and Data Sources as Grid Services," *Junwei Cao (Ed.), Cyberinfrastructure Technologies and Applications, Nova Science Publishers, New York, USA*, 2009.

[15] S. Benkner, A. Arbona, G. Berti, A. Chiarini, R. Dunlop, G. Engelbrecht, A. F. Frangi, C. M. Friedrich, S. Hanser, P. Hasselmeyer, R. D. Hose, J. Iavindrasana, M. Köhler, L. L. Iacono, G. Lonsdale, R. Meyer, B. Moore, H. Rajasekaran, P. E. Summers, A. Wöhrer, and S. Wood, "@neurist: Infrastructure for advanced disease management through integration of heterogeneous data, computing, and complex processing services," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, no. 6, pp. 1365–1377, November 2010.

[16] M. Antonioletti, M. Atkinson, R. Baxter, A. Borley, C. Hong, P. Neil, B. Collins, N. Hardman, A. C. Hume, A. Knox, M. Jackson, A. Krause, S. Laws, J. Magowan, N. W. Paton, D. Pearson, T. Sugden, P. Watson, and M. Westhead, "The design and implementation of grid database services in ogsa-dai: Research articles," *Concurrency and Computation : Practice and Experience*, vol. 17, no. 2-4, pp. 357–376, 2005.

[17] M. N. Alpdemir, A. Mukherjee, A. Gounaris, N. W. Paton, P. Watson, A. A. Fernandes, and D. J. Fitzgerald, "Ogsa-dqp: A service for distributed querying on the grid," in *Advances in Database Technology - EDBT 2004*, ser. Lecture Notes in Computer Science, E. Bertino, S. Christodoulakis, D. Plexousakis, V. Christophides, M. Koubarakis, K. Böhm, and E. Ferrari, Eds. Springer Berlin / Heidelberg, 2004, vol. 2992, pp. 3923–3923.

[18] A. Wöhrer, P. Brezany, and A. M. Tjoa, "Novel mediator architectures for grid information systems," *Future Generation Computer Systems*, vol. 21, no. 1, pp. 107 – 114, 2005.

[19] D. Teodoro, R. Choquet, E. Pasche, J. Gobeill, C. Daniel, P. Ruch, and C. Lovis, "Biomedical data management: a proposal framework," in *Stud Health Technol Inform*, 2009.

[20] A. Branson, T. Hauer, R. McClatchey, D. Rogulin, and J. Shamdasani, "A data model for integrating heterogeneous medical data in the health-e-child project," *CoRR*, vol. abs/0812.2874, 2008.

[21] G. A. Komatsoulis, D. B. Warzel, F. W. Hartel, K. Shanbhag, R. Chilukuri, G. Fragoso, S. de Coronado, D. M. Reeves, J. B. Hadfield, C. Ludet, and P. A. Covitz, "cacore version 3: Implementation of a model driven, service-oriented architecture for semantic interoperability," *Journal of Biomedical Informatics*, vol. 41, no. 1, pp. 106 – 123, 2008.