TOWARDS DATA AND DATA QUALITY MANAGEMENT FOR LARGE SCALE HEALTHCARE SIMULATIONS
[Position Paper]

Philipp Baumgärtel and Richard Lenz
Chair for Computer Science 6 (Data Management), Friedrich-Alexander University of Erlangen-Nuremberg, Germany
On behalf of the ProHTA Research Group
philipp.baumgaertel@cs.fau.de, richard.lenz@cs.fau.de

Keywords: Simulation Data Management : Knowledge Management : Ontologies : Healthcare

Abstract: The approach of ProHTA (Prospective Health Technology Assessment) is to understand the impact of medical processes and technologies as early as possible. Therefore, simulation techniques are utilized to estimate the effects of innovative health technologies and find potentials of efficiency enhancement within the supply chain of healthcare. Data management for healthcare simulations is required as heterogeneous data is needed both as simulation input data and for validation purposes. The main problem is the heterogeneity of the data and the initially unknown and continuously changing demands of the simulation. Also, data quality considerations are necessary to quantify the reliability of simulation output. A solution has to consider all of these aspects and must be extensible to cope with changing requirements. As the structure of the data is not known in advance, a generic database schema is required. This paper proposes an approach to store heterogeneous statistical data in an RDF-triplestore. Semantic annotations based on conceptual models are utilized to describe the datasets. Additionally, a special query language helps loading the data into the simulation. The feasibility of the approach has been demonstrated in a prototype implementation. We discuss the benefits of this approach as well as remaining challenges and issues.

1 INTRODUCTION

The main goal of ProHTA (Prospective Health Technology Assessment) is to simulate medical processes to gain information about the impact of diverse new health technologies on healthcare. Therefore, a modular simulation framework has to be designed to answer questions about different new medical products.

Besides the problems of simulation modeling, validation and optimization, simulation data management is required. Skoogh et al. (Skoogh et al., 2010) claim that the input data management process consumes about 31% of the time of a simulation study. They argue that in most cases the data is collected manually for each simulation study. Robertson and Perera (Robertson and Perera, 2002) conducted a survey showing that 60% of the polled simulation practitioners manually input the data to the simulation model.

ProHTA shall become a framework to be used to answer different questions in the same domain. Hence, the reusability of simulation model components and input data is important. The main data management problem is to store heterogeneous data in such a way as to ensure its reusability. Typical data sources contain preaggegated data, like e.g. demographic data, healthcare statistics, geographic data etc. These data are to be provided, both to feed the heathcare simulation with realistic parameters and to validate the simulation. To be able to cope with rapidly growing data sets and new unknown data sources, we propose an approach to store arbitrary statistical data without previously fixing its semantics in a database schema. By semantically annotating the stored data, we are able to search for already integrated datasets for reuse.

Another major problem is data quality. Because decisions may be based on the simulation output, its reliability is important. Therefore, the simulation models have to be validated and the quality of the input data has to be quantified. Data provenance is important for simulation studies to be repeatable and to determine data quality (Stonebraker et al., 2009). Additionally, storing the inherent uncertainty of scientific data is important to quantify the quality of simulation output (Stonebraker et al., 2009).

HEALTHINF 2012 - Proceedings of the International Conference on Health Informatics (DOI: 10.5220/0003871602750280)

There are two main questions regarding simulation input data management:

1. How can heterogeneous statistical data be stored and queried to be reusable in many different simulation studies?
2. How can data quality be quantified to estimate the reliability of the simulation output?

In this paper, we present our approach to store simulation input data in an RDF-triplestore. Additionally, we outline a simple query language for statistical simulation input data. After the discussion of data quality issues and related work on simulation input data management, we will conclude with a summary and a perspective on future work on this subject.

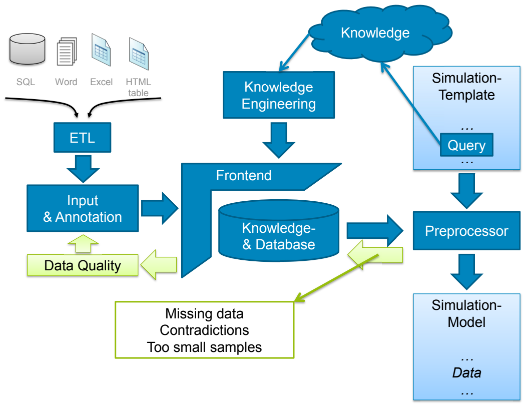

Figure 1: Data management concept for healthcare simulations

2 DATA MANAGEMENT FOR LARGE SCALE HEALTHCARE SIMULATIONS

Figure 1 depicts our basic data management concept for healthcare simulations. One of the main problems is the heterogeneity of the data sources. Therefore, a manual ETL process has to be applied to integrate the data in a central storage. Also, knowledge management is important to organize the different datasets.

To be independent from simulation languages and tools, our data management concept uses a preprocessor. A simulation template containing the regular simulation program and queries is processed. The queries in the simulation templates are replaced by real data. Then the resulting simulation containing the data can be processed by the simulation tool.

The preprocessor also checks data quality constraints and provides feedback to the user. It also encourages the user to improve the quality of the simulation's input data.

At the core of our concept is the storage component for data and knowledge. Basically, three approaches exist to store heterogeneous data with unknown or changing semantic constraints. One possible solution is a schema-free approach. However, data in a schema-free storage can not be reused or even be queried or processed easily. An adaptable schema would solve this problem. However, when the schema is adapted, we will need to adapt the applications using the data as well. Our proposed solution is a generic schema flexible enough to cope with changing requirements and heterogeneity but also structured enough to support querying and reusing the data.

A generic relational approach like EAV (Nadkarni et al., 1999) can be utilized to store information about entities with an arbitrary number of attributes. This technique has been adapted for various purposes including the storage of structured document contents (Lenz et al., 2002). The price for flexibility is a loss of semantic control at the database level. In addition, queries on an EAV-schema tend to be more complex than traditional queries. Attempts to regain semantic control by the use of additional user defined metadata tables even aggravate the problem of query complexity (Nadkarni et al., 1999). Despite these trade offs, the usefulness of the EAV approach for certain purposes is out of question.

In our context, the statistical input data is typically multidimensional rather than document based. However, because of the heterogeneity of the data, a conventional data warehouse approach is not suitable. Also, at the current state of ProHTA, the requirements for the multidimensional storage are not predictable. Our attempts to design a relational schema to store multidimensional data with arbitrary dimensions and a flexible number of attributes resulted in EAV-like relations containing triples. Because of the drawbacks of EAV, we decided to choose RDF (Lassila et al., 1999), being inherently triple-based, to store the simulation input data and additional metadata.

2.1 Storing Heterogeneous Multdimensional Data in RDF

We compared the most prominent approaches to describe multidimensional data in RDF in Table 1.

Modeling dimensions as properties instead of classes results in less triples than storing dimensions as separate instances, but is also less flexible. Because of the need to annotate dimensions with further information, a class based approach is required in our scenario.

Only the approach of Kurze et al. (Kurze et al., 2010) supports classification hierarchies in the dimen-

Table 1: Comparison of different ontologies for multidimensional data

Ontology	Class based dimensions	Hierarchies	Summarizability	Unit	Multi-measure observations
(Cyganiak et al., 2010)	No	No	No	Yes	Yes
(Hausenblas et al., 2009)	Yes	No	No	No	No
(Niemi et al., 2007)	No	No	No	No	Yes
(Niemi and Niinimäki, 2010)	No	No	Yes	Yes	Yes
(Kurze et al., 2010)	Yes	Yes	No	No	Yes
Requirements of ProHTA	Yes	Yes	Yes	Yes	Yes

sions. Unfortunately they do not describe their approach in sufficient detail.

Summarizability information (Lenz and Shoshani, 1997) is only considered by Niemi and Niinimäki (Niemi and Niinimäki, 2010). This is important for automatic aggregation. Other important aspects are the unit of the measured data and the support for multi-measure observations.

To our knowledge, no well documented approach to store multidimensional data in RDF fulfilling all our requirements exists. Therefore, we developed our own RDF-schema.

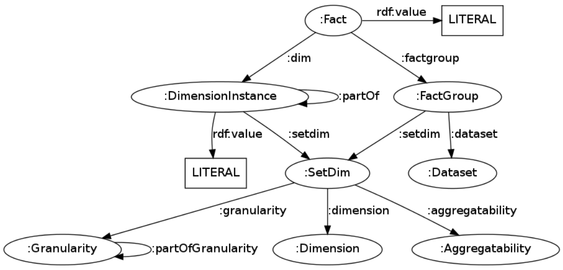

Figure 2: RDF-schema to store multidimensional data

A simplified version of our data model is depicted in Figure 2. This schema was designed to support efficient querying and to avoid redundancy. There are facts linked to different observations, although in Figure 2 there is only one value (rdf:value) depicted. Facts can be grouped together and one dataset can contain multiple groups of facts. There are dimensions like age, time or gender. A specific fact group has different dimensions in a specific granularity like day, month or year. Also, the information how to aggregate along a given dimension is stored for each group of facts. At the moment, this is the only summarizability information we store. These informations are stored using the artificial class :SetDim. Facts are identified by the instances of one dimension like a specific day in the time dimension. The hierarchy of dimension instances is stored using the :partOf property. Additionally, the hierarchy of granularities is stored using the :partOfGranularity property. This is not redundant, because the hierarchical dependencies between granularities have to be stored even if no dimension instances in these granularities exist. Additionally, the connection between dimension instances in different granularities can not be derived from the hierarchy of granularities. However, inconsistencies between these two hierarchies are possible and have to be prevented.

We also store information about the unit of observations in one group of facts. Each unit is linked to a base unit and the conversion factors between a unit and it's base unit are stored. However, this is not depicted in Figure 2.

2.2 Knowledge Management

Currently, the different datasets are only identified by their name. When the simulation models grow in size and detail, the problem of finding the appropriate dataset in the RDF-triplestore will arise. Therefore, it is necessary to store context information about datasets. Hence, detailed semantic descriptions of datasets are required.

In our simulation project, we are developing conceptual models as a first step towards executable simulation models. These conceptual models can be formalized using the RDF ontology we are currently developing. Then, data and simulation models are able to reference the formalized conceptual models and data sets can be queried using terms from the conceptual models.

The main problem is that many different conceptual models are needed for our simulation project. There are, for example, high level models representing stocks and flows and more detailed models representing individuals and decisions.

Figure 3 depicts our idea to solve this problem. A fixed meta language is used to describe different modeling languages. For example, we can describe a language to describe stock and flow models.

These modeling languages are then used to de-

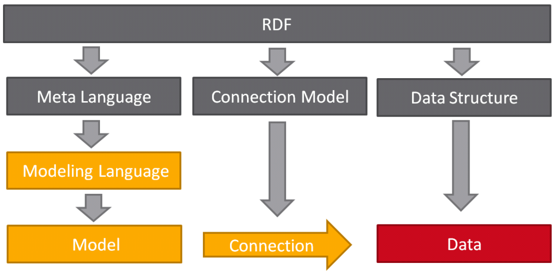

Figure 3: Describing conceptual models in RDF

scribe individual conceptual models. The data needed to execute our simulations can be described using a connection model to create links between data and conceptual models. The structure to actually store the data has been explained in Section 2.1.

This approach is currently under development.

3 A QUERY LANGUAGE FOR HEALTHCARE SIMULATIONS

Another benefit from simulation data management is the independence between data collection and model building. Once the data is stored, the simulation modeler is exempt from the task to manually input the data to the simulation model. The task of the simulation modeler is now to query the data from the data storage component. To load data into the simulation two kinds of information are necessary:

1. What dataset should be loaded?
2. In which form should the data be loaded? (E.g. dimensions, granularity, unit, ...)

The data is semantically annotated in RDF utilizing conceptual models and our schema for multidimensional data. Therefore, both the first and second question could be answered in SPARQL (Prud'hommeaux and Seaborne, 2008). However, SPARQL queries selecting appropriate data items would be very complex because our RDF schema contains additional meta data. Another problem is that aggregation is not part of the SPARQL standard. It would be much more convenient and powerful to use a query language for multidimensional data like MDX (Multidimensional Expressions). We are currently developing a simple query language specifically designed for statistical simulation input data . That way, the queries are independent from the actual multidimensional RDF structure.

For example, a query to get the number of men aged between 50 and 60 years in a population depending on age and time could be expressed as:

```
select cube<One> (Time<Year>,
                  Age<Year> = [50-60],
                  Gender<MW> = "M")
from Population;
```

One additional information contained in this query is the desired unit of the observations. In this example, the unit is simply “One”.

At the moment datasets are selected by name. In the future the connection to conceptual models will be utilized to find datasets.

If more information than just one value per fact is stored, it can be queried by adding arbitrary SPARQL statements to our query language. That way arbitrary information with multidimensional structure can be stored and queried. For example, the time of diagnosis and treatment steps in a hospital depending on age and gender could be stored in one data cube.

```
select ?ci cube<One> (Time<Year>,
                  Age<Year> = [50-60],
                  Gender<MW> = "M")
from Population
with {
    ?fact data:confidence_interval ?ci .
};
```

Because our query language adds unit conversion and automatic aggregation to SPARQL, we can not simply translate queries. To process a query in our language, it is checked whether an appropriate dataset with sufficient data quality exists in the data storage. The factor to convert the observations to the desired unit is calculated. Then, the dataset is aggregated to the desired dimensions and granularities and the resulting dataset is stored as a new group of facts for provenance reasons. After that, the remaining query processing is done by translating the query to an equivalent SPARQL query.

4 DATA QUALITY CONSIDERATIONS

As previously mentioned, measurability of data quality contributes to the success of ProHTA. Only simulation results with quantifiable reliability are useful.

Accuracy is the most prominent data quality dimension, however Wang and Strong (Wang and Strong, 1996) listed other types of data quality as well. Besides the quality of the input data, other aspects influence the quality of the simulation output. High accuracy and appropriate granularity of the simulation model are required for precise results. Also, the characteristics of the simulation model's error propagation have influence on the output's quality.

For example, Cheng and Holland (Cheng and Holland, 2004) developed an approach to calculate confidence intervals for simulation output. This method is concerned with the variability arising from the use of random numbers in the simulation and the uncertainty of the parameters.

In order to continuously improve data quality we need a methodological approach to data quality management (Batini et al., 2009). Firstly, the necessary data quality dimensions have to be identified. Then, the data quality has to be converted to a statistical measure (e.g. a confidence interval). The simulation model's error propagation characteristics have to be evaluated. Additionally, the accuracy of the simulation model itself has to be estimated by a validation procedure. Finally, the statistical measure has to be propagated through the simulation model. That way the accuracy of the simulation output can be calculated.

Knowledge engineering is applied to annotate the stored data. This can also be useful for data quality considerations. Fürber and Hepp (Fürber and Hepp, 2010) proposed an approach to correct wrong values using semantically annotated reference data. They provided SPARQL queries for identifying missing or illegal values. Another example in our context would be data from a medical study conducted only in some hospitals. Then, an ontology describing hospitals, cities and populations could be utilized to estimate the generality of this data.

5 RELATED WORK

There are different technical approaches to support reusability of simulation data. Gowri (Gowri, 2001) presented EnerXML, an XML schema to enable interoperability between different energy simulations. Bengtsson et al. (Bengtsson et al., 2009) proposed the Generic Data Management Tool. It stores input data for discrete event manufacturing simulations according to the Core Manufacturing Simulation Data (CMSD) specification. Boulonne et al. (Boulonne et al., 2010) extended the Generic Data Management Tool by a Resource Information Management component. This component enables the reuse of resource information by generating standard CMSD files.

Another approach to structured simulation data management are simulation workflows. Reimann et al. (Reimann et al., 2011) introduced SIMPL – a framework for data provisioning for simulation workflows. This framework supports the ETL-process (extract, transform and load) for simulation data. Data is stored as XML, which is flexible enough for heterogeneous data. However, the problem of designing a schema to support reusability of input data remains unsolved.

SciDB (Rogers et al., 2010) is a project aimed at scientific data management. The main goal of this project is to store large multidimensional array data and to support efficient data processing. In contrast to SciDB, our main problem is not to handle very large array data, but to handle many small multidimensional data sets in different granularities with potentially complex datatypes.

A project by Ainsworth et al. (Ainsworth et al., 2011) aims at simulating healthcare policy interventions in a generic way. Their data management component simply uses the NHibernate Framework to store simulation input data. This approach does not solve the problem of data heterogeneity. Additionally, they do not concern data quality and data reusability issues.

Zhang et al. (Zhang et al., 2011) introduced SciQL, a query language for scientific applications. Like SciDB's query language, SciQL is designed to query multidimensional arrays, but not for data warehouses with hierarchical dimensions.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we clarified the importance of data and data quality management in simulation studies. Then, we proposed a simulation data management approach using an RDF-triplestore and described how to query it. Afterwards we discussed data quality challenges and issues for simulations. Finally, we discussed related work.

There are three main benefits from our approach. Our RDF schema is flexible enough to cope with the changing demands of the simulation. By semantic annotations utilizing conceptual models and metadata, the datasets are also structured enough to be reusable. The query language, we are developing, helps to load data into the simulation independently from the actual multidimensional RDF data structure. Additionally, the stored knowledge and metadata can be used to control and improve data quality.

We validated our approach with a prototypical implementation of our framework (Figure 1). In future work, we will identify the data quality requirements of a ProHTA simulation study in detail. Also, we will study how to control and improve data quality by using stored knowledge. Additionally, we will refine our approach to store conceptual models and to utilize them to annotate stored datasets. Finally, the query language for statistical simulation input data will be improved.

ACKNOWLEDGEMENTS

This project is supported by the German Federal Ministry of Education and Research (BMBF), project grant No. 01EX1013B.

REFERENCES

Ainsworth, J. D., Carruthers, E., Couch, P., Green, N., O'Flaherty, M., Sperrin, M., Williams, R., Asghar, Z., Capewell, S., and Buchan, I. E. (2011). Impact: A generic tool for modelling and simulating public health policy. *Methods of Information in Medicine*, 5:454–463.

Batini, C., Cappiello, C., Francalanci, C., and Maurino, A. (2009). Methodologies for data quality assessment and improvement. *ACM Comput. Surv.*, 41:16:1–16:52.

Bengtsson, N., Shao, G., Johansson, B., Lee, Y., Leong, S., Skoogh, A., and Mclean, C. (2009). Input data management methodology for discrete event simulation. In *Winter Simulation Conference (WSC), Proceedings of the 2009*, pages 1335 –1344.

Boulonne, A., Johansson, B., Skoogh, A., and Aufenanger, M. (2010). Simulation data architecture for sustainable development. In *Proceedings of the 2010 Winter Simulation Conference*.

Cheng, R. C. H. and Holland, W. (2004). Calculation of confidence intervals for simulation output. *ACM Trans. Model. Comput. Simul.*, 14:344–362.

Cyganiak, R., Reynolds, D., and Tennison, J. (2010). The rdf data cube vocabulary. `http://publishing-statistical-data.googlecode.com/svn/trunk/specs/src/main/html/cube.html`.

Fürber, C. and Hepp, M. (2010). Using semantic web resources for data quality management. In *Proceedings of the 17th international conference on Knowledge engineering and management by the masses*, EKAW'10, pages 211–225, Berlin, Heidelberg. Springer-Verlag.

Gowri, K. (2001). Enerxml - a schema for representing energy simulation data. In *Proceedings of the Seventh International IBPSA Conference*.

Hausenblas, M., Halb, W., Raimond, Y., Feigenbaum, L., and Ayers, D. (2009). Scovo: Using statistics on the web of data. In *The Semantic Web: Research and Applications*, volume 5554 of *Lecture Notes in Computer Science*, pages 708–722. Springer Berlin / Heidelberg.

Kurze, C., Gluchowski, P., and Bohringer, M. (2010). Towards an ontology of multidimensional data structures for analytical purposes. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pages 1 –10.

Lassila, O., Swick, R. R., Wide, W., and Consortium, W. (1999). Resource description framework (rdf) model and syntax specification. `http://www.w3.org/TR/1999/REC-rdf-syntax-19990222`.

Lenz, H.-J. and Shoshani, A. (1997). Summarizability in olap and statistical data bases. In *Scientific and Statistical Database Management, 1997. Proceedings., Ninth International Conference on*, pages 132 –143.

Lenz, R., Elstner, T., Siegele, H., and Kuhn, K. A. (2002). A practical approach to process support in health information systems. *Journal of the American Medical Informatics Association*, 9(6):571–585.

Nadkarni, P. M., Marenco, L., Chen, R., Skoufos, E., Shepherd, G., and Miller, P. (1999). Organization of heterogeneous scientific data using the eav/cr representation. *Journal of the American Medical Informatics Association*, 6(6):478–493.

Niemi, T. and Niinimäki, M. (2010). Ontologies and summarizability in olap. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, pages 1349–1353, New York, NY, USA. ACM.

Niemi, T., Toivonen, S., Niinimaki, M., and Nummenmaa, J. (2007). Ontologies with semantic web/grid in data integration for olap. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 3:25–49.

Prud'hommeaux, E. and Seaborne, A. (2008). Sparql query language for rdf. `http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/`.

Reimann, P., Reiter, M., Schwarz, H., Karastoyanova, D., and Leymann, F. (2011). Simpl - a framework for accessing external data in simulation workflows. In *Datenbanksysteme fr Business, Technologie und Web (BTW)*.

Robertson, N. and Perera, T. (2002). Automated data collection for simulation? *Simulation Practice and Theory*, 9(6-8):349 – 364.

Rogers, J., Simakov, R., Soroush, E., Velikhov, P., Balazinska, M., DeWitt, D., Heath, B., Maier, D., Madden, S., Patel, J., Stonebraker, M., Zdonik, S., Smirnov, A., Knizhnik, K., and Brown, P. G. (2010). Overview of scidb, large scale array storage, processing and analysis. In *Proceedings of the SIGMOD'10*.

Skoogh, A., Michaloski, J., and Bengtsson, N. (2010). Towards continuously updated simulation models: Combingin automated raw data collection and automated data processing. In *Proceedings of the 2010 Winter Simulation Conference*.

Stonebraker, M., Becla, J., DeWitt, D., Lim, K.-T., Maier, D., Ratzesberger, O., and Zdonik, S. (2009). Requirements for science data bases and scidb. In *Proceedings of the CIDR 2009 Conference*.

Wang, R. Y. and Strong, D. M. (1996). Beyond accuracy: what data quality means to data consumers. *J. Manage. Inf. Syst.*, 12:5–33.

Zhang, Y., Kersten, M., Ivanova, M., and Nes, N. (2011). Sciql, bridging the gap between science and relational dbms. In *Proceedings of the IDEAS11*.