SEMANTIC DATA INTEGRATION FOR PROCESS ENGINEERING DESIGN DATA

Andreas Wiesner, Jan Morbach and Wolfgang Marquardt
AVT-Process Systems Engineering, RWTH Aachen University, Templergraben 55, 52062-Aachen, Germany

Keywords: Data integration, data consolidation, ontology, XML technologies, engineering design data.

Abstract: During the design phase of a chemical plant, information is typically created by various software tools and stored in different documents and databases. Unfortunately, the further processing of the data is often hindered by the structural, syntactic and semantic heterogeneities of the data sources. In fact the merging and consolidation of the data becomes virtually prohibitive when exclusively conventional database technologies are employed. Therefore, XML technologies as well as specific domain ontologies are increasingly applied in the context of data integration. Hence, this contribution gives an outline on an ongoing research project at the authors' institute, which aims at the development of a prototypical software tool, which exploits the benefits of semantic as well as XML technologies, for the integration and consolidation of design data. Both, ontology and software development is performed in close cooperation with partners from the chemical and software industries to ensure their compliance with the requirements of industrial practice.

1 INTRODUCTION

In the course of a chemical plant design project, information is typically created by disparate tools and stored in different locations and formats (e.g. technical documents, CAE systems and simulation files). However, before further processing, the scattered information has to be merged and consolidated. Unfortunately, in practice, data integration projects are hindered by the inherent heterogeneities of the underlying sources. (Embury et al., 2001). As a result the lack of interoperability between the tools and data stores causes a significant overhead for the designers as they have to spend considerable time on the re-entering of data, the manual consolidation of overlapping data sets, and the search for information (Galaher et al., 2004). In order to overcome the aforementioned heterogeneities, XML is increasingly applied for data exchange purposes, ultimately becoming a standard for data interchange between software tools (Klein, 2002). Hence, various XML-based applications for data exchange in the field of chemical engineering already exist or are currently under development such as CAEX (Fedai and Draht, 2004), XMpLant (Noumonon, 2006) or PlantXML (Anhäuser et al., 2004).

Thus, at least syntactic and schematic heterogeneities can be resolved conveniently between distributed data sources by means of the XML format. However, XML and its schemas do not express semantics (Cruz et al., 2004), such that semantic incompatibility between different XML sources is inevitable.

For the integration of data from several different sources, particularly for engineering data, correct assumptions about the meaning of certain elements are crucial for the successful information retrieval and consolidation. In other words, where XML sources are presented without an explicit agreement on the semantics of certain tags and document structures, the task of the correct interpretation of the data is still an issue (Erdmann and Studer, 2001).

To remedy this problem, a semantic annotation, also referred to as "semantic lifting" of XML documents is necessary. Unfortunately, the semantics assumed by a particular source are rarely documented, and there is no explicit representation of a data source's semantics, in the way that a schema provides a representation of the data structure. Hence, the important link missing at this point is the connection between the structured information stored in the XML document and the particular domain knowledge, which relates meaning to the stored information within the context. To that

Wiesner A., Morbach J. and Marquardt W. (2008).
SEMANTIC DATA INTEGRATION FOR PROCESS ENGINEERING DESIGN DATA.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - DISI*, pages 190-195
DOI: 10.5220/0001692901900195

end, ontologies have gained popularity as a convenient means for the representation of domain knowledge.

An ontology is an explicit specification of a conceptualization, typically involving classes, their relations and axioms for clarifying the intended semantics (Uschold and Grüninger, 1996). It basically constitutes a structured framework for the storage of information and knowledge. Often the ontologies are linked with the term semantic technology. By semantic technologies, software systems are meant that use ontologies as internal data models. The ontology OntoCAPE (Morbach et al. 2007, Morbach and Marquardt, 2008) was explicitly defined for the domain of Computer-Aided Process Engineering and thus particularly applicable to the integration of design data in chemical engineering.

This contribution reports on the ongoing research project "Ontology-based integration and management of distributed design data" at the authors' institute, that address the aforementioned semantic heterogeneities between documents containing process engineering design data due to the lack of interoperability between software tools. The project's aim is to develop an ontology-based software prototype, incorporating OntoCAPE, for the integration and reconciliation of design data which are available in the XML data format, from distributed information sources. The project is run in cooperation with partners from the chemical and software industries. This paper, however, will particularly emphasis the conceptual design and implementation of the novel software tool.

The remainder of the paper is organized as follows: Section 2 introduces the concept of "semantic lifting" and gives a brief overview on the OntoCAPE ontology. In Section 3 the conceptual design and the implementation of the current research project are introduced. Finally, Section 4 concludes the contribution by summarizing the achievements so far.

2 PREREQUISITE FOR SEMANTIC DATA INTEGRATION

The main purpose of XML is to provide a mechanism that can be used to mark-up and structure documents. This allows machines to identify pieces of data in a document by their label. However, these labels themselves do not bear any meaning with them. Also, it is a common misconception that XML schema documents can be used to add meaning to XML documents (Klein, 2002). The goal of XML schema mainly is to provide structuring prescriptions, e.g. the feature to build hierarchies of element types, which, however, do not contain conceptual knowledge, but only functions as a syntactical shortcut to allow reuse of complex definitions.

2.1 Semantic Lifting

To associate some meaning with XML documents, it is necessary to relate the labels with something that carries meaning. Classes and properties in ontologies are suitable for that purpose, because ontologies formally specify the understanding of certain topics in a particular application domain. A naive way of establishing the relation between an ontology and the XML document would be a simple matching of associated labels from a XML document syntactically with the names of classes and properties in the ontology. But meaning can rarely be assigned by a simple mapping from symbols to objects since the role of the data implicitly indicated by the context, e.g. the nested structure of a document, is not clearly captured this way. Accordingly, for a reliable data consolidation it is crucial to unambiguously interpret the data including its context. To that end, a substantial description of the XML document's contents ("semantic lifting") by means of an ontology has to be provided in order to undertake a proper semantic data integration.

The proposed "semantic lifting" follows a two step approach: step (1) is to lift the XML schema to the level of an ontology, i.e. a skeleton "schema ontology" is created which incorporates only the hierarchical information from the schema expressed in an ontology language. Step (2) establishes further relations between concepts and attributes in order to clarify context information which goes beyond simple hierarchical interrelations essentially leading to a "document ontology". So far, however, most of the modeling in step (2) has to be done manually.

2.2 Prototype Architecture

This software prototype basically represents a mediation layer, which is placed between the user and the data sources. The tool follows the *local-as-view* approach (Levy, 2001), i.e. the architecture consists of a global schema (the domain ontology), a source schema (document ontology) and mappings as proposed by Lenzerini (2002). A schematic representation is given in Fig.1. The user interacts with the tool by querying the global schema, which constitutes a virtual representation of the data existing in the data sources. The tool then carries out

the task of dealing with the sources to retrieve the information satisfying the user's request. Moreover, the source schema provides an internal representation of the data at the sources. Finally, the relationships existing between the entities of the global and source schema are represented by mappings. By means of these mappings the aforementioned missing link is established such that the particular domain knowledge relates meaning to the source schema. The interaction between the software prototype and the XML sources is realized by a specific, bidirectional converter. Therefore, the new approach basically considers the tool to be layered on top of an existing XML-data-exchange architecture, in the sense of "publish and subscribe", for serialization.

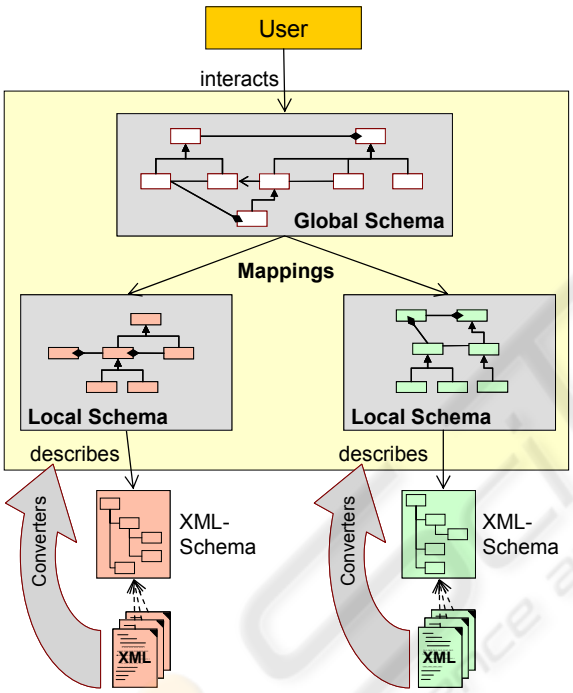

Figure 1: Schematic representation of the integration tool.

2.3 Domain Ontology OntoCAPE

To enable an adequate description of the contents of the XML files a comprehensive information model of the design process of a chemical plant has to be provided. Such particular domain knowledge is captured by the OntoCAPE ontology. The formal ontology OntoCAPE captures consensual knowledge of the application domain in such a way that it can be reused and shared across software systems. It specifies the meaning of the vocabulary terms and constrains its interrelations (and its possible uses) by means of axiomatic definitions. Then, specialized software components (so-called inference engines or reasoners) can be applied to interpret and reason about the data.

OntoCAPE has been designed for use with different types of CAPE tools that support such diverse tasks as mathematical modeling (Braunschweig et al. 2002, Yang and Marquardt, 2004), knowledge management (Brandt et al. 2008), and data integration (Morbach and Marquardt, 2008). An extensive documentation of OntoCAPE publicity is available at (OntoCAPE, 2007).

OntoCAPE is organized through three types of structural elements: layers, modules, and partial models (cf. Fig. 2).

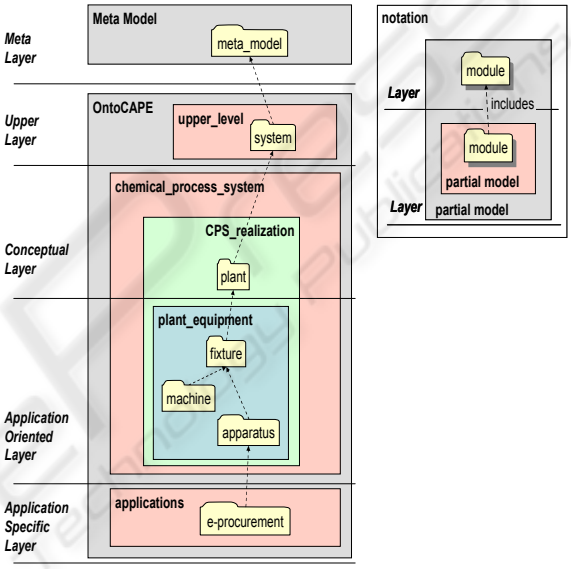

Figure 2: A detail of OntoCAPE demonstrating the overall structure of the ontology.

The layers subdivide OntoCAPE into different levels of abstraction, thus separating general knowledge from knowledge about particular domains and applications. The topmost *Meta Layer* is the most abstract one. It holds a Meta Model which introduces fundamental modeling concepts and states the design guidelines for the construction of the actual ontology. Next, the *Upper Layer* of OntoCAPE defines the principles of general systems theory according to which the ontology is organized. On the subjacent *Conceptual Layer*, a conceptual model of the CAPE domain is established, which covers such different areas as unit operations, equipment and machinery, materials and their thermophysical properties, chemical process behavior, modeling and simulation, and others. The two bottommost layers refine the conceptual model by adding classes and relations required for the practical application of the ontology: The *Application-Oriented Layer* generically extends the ontology towards certain application areas, whereas

the *Application-Specific Layer* provides specialized classes and relations for concrete applications.

A module assembles a number of interrelated classes, relations, and axioms, which jointly conceptualize a particular topic (e.g., the module 'plant' provides a conceptualization of chemical plants). The boundaries of a module are chosen such that the module can be designed, adapted, and reused to some extent independently from other parts of an ontology (Stuckenschmidt and Klein, 2003). Modules addressing closely related topics are grouped into a common partial model (e.g., the partial model 'plant_equipment' clusters the thematically related modules 'fixture', 'apparatus', and 'machine').

The modules presented in Fig. 1, e.g. 'system', 'plant' etc., comprise the basic principles for the particular domain knowledge required in the project. For a comprehensive description of the modules, we refer to (OntoCAPE, 2007)

3 PROJECT OUTLINE

The project aims at developing a data integration prototype to address the problem of semantic interoperability between different engineering documents generated in a chemical plant design process. To that end, the tool incorporates ontology-based information reconciliation of data expressed in the XML format. Ultimately, the tool is intended to support the designers by providing an integrated view on the relevant project data and by enabling efficient data retrieval. A further functionality of the tool is the automatic detection of design errors: For example, a typical design error would be the interconnection of flanges with inconsistent internal diameters. As the main objectives, the integration tool must assemble, integrate and consolidate the relevant information required in the design process.

3.1 Method

The software prototype follows a two-step approach for data integration: In step (1), the current and relevant information is identified, extracted, and prepared for further processing; in step (2), the information is integrated, and their inconsistencies are reconciled. As prerequisite, both steps require a "semantic lifting" of the information stored in the XML files. The integration steps are carried out in the comprehensive information base (CIB). Fig.3 gives a schematic representation.

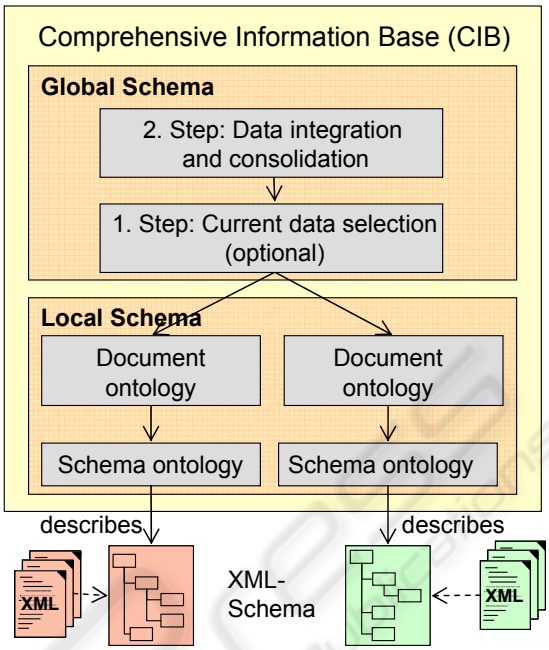

Figure 3: Schematic representation of the CIB.

For testing purposes, the integration tool currently employs PlantXML files, which is a data exchange format realized via XML files that comply with a company-internal standard schema. PlantXML is the existing in-house solution for the information exchange between application tools which has been implemented by the engineering department of our project partner Evonik Degussa (Anhäuser et al., 2004): PlantXML defines specific XML schemata for the different phases and crafts of a design project: XML-EQP for the design of machines and apparatuses, XML-EMR for the design of instruments and control systems, XML-RLT for piping engineering, and XML-SiAr for the design of fittings and safety valves. However, the novel integration tool is designed in such a way, that it can process any data in the XML format as long as it complies with an available XML schema.

In step (1), the relevant information items from each of the scattered sources in terms of PlantXML files must be identified, extracted, and assembled in the CIB. PlantXML's organization and structure is accommodated to the project designers' workflow and supports parallel and distributed workmanship according to the complex workflow of a design project, e.g. split of work, concurrent engineering, and distributed engineering. Correspondingly, in the course of a project several instances of the different PlantXML schema are generated. In other words, different versions of identical real world items exist. Hence, no stringent versioning of the data items is possible according to the complex workflow of

identical real world objects in different crafts. Thus, a versioning of the data items is required, i.e., the current information of each source has to be determined. Furthermore, redundant information has to be detected and rejected. At the end of step (1) the designer may choose one of the two options: the pre-processed data can either be reconverted to the PlantXML format and thus be integrated in the existing workflow, or one obtains a pre-processed data set for further processing in step (2). Note that step (1) is optional. Considering an alternative XML data set with an existing versioning procedure, the actual and relevant information may be extracted to the CIB for a direct processing at step (2).

Step (2) performs the actual integration, which essentially requires the merging of the current information from each source and the check for inconsistent information from the different sources. For the latter, the global schema (domain ontology based on OntoCAPE) established within the CIB provides the necessary vocabulary in terms of classes and relations, and defines the feasible interrelations between the vocabulary terms. This way, it can be checked if the semantic representation of the merged information complies with the feasible interrelations defined in the domain ontology. As a result, potential design errors and inconsistencies can be detected and reported to the designer. A typical example for step (2) is given in Fig. 4, where the connection of a vessel extracted from the XML-EQP schema to a pipe taken from the XML-EMR schema is validated against the domain ontology.

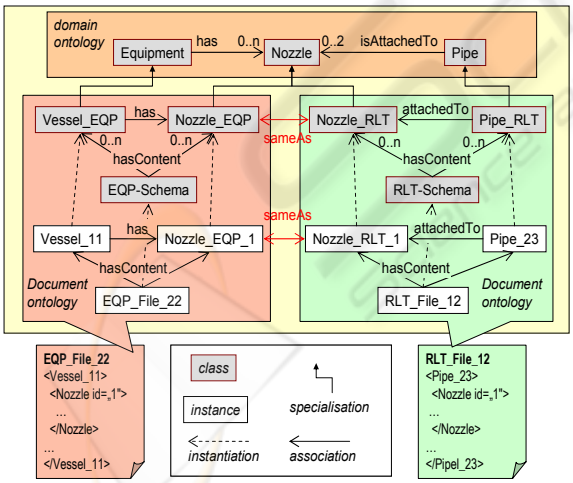

Figure 4: Application example of the CIB.

3.2 Implementation

The core of the implementation is the CIB as mentioned before. Essentially, we intend to use semantic technologies for the realization of the prototypical software tool as far as possible.

An advantage of ontology-based systems over conventional database technology is the possibility to partially automate the information integration process. The inference mechanism of deduction (i.e. execution of production rules) is especially applicable for this aim. Deduction is particularly useful for merging and consolidating of distributed information (Maier et al., 2003) and thus decided to employ a deductive language (and a compatible inference engine) for the integration tool.

However, some of the requirements on the CIB might still be achieved more conveniently by conventional database technologies. As an example consider mass data which do not require a semantic enrichment for consolidation purposes.

Thus, an implementation basis has to be chosen which fulfils the requirements for both technologies equally well. Accordingly, the development system OntoStudio (OntoStudio, 2007), which has been developed by the project partner ontoprise, serves as the implementation basis. Unlike most other ontology-based systems available today, OntoStudio is scalable and thus suitable for processing of large data as it is presumed in this project.

It relies on the deductive ontology language F-Logic (Kifer et al., 1995), which allows the definition of rules for integration and mapping purposes, and the formulation of queries. These rules represent declarative knowledge in the form “if A then B”, where A and B are statements about the extracted information expressed by means of ontological terms. This approach is more intuitive and less error-prone than conventional database integration, especially in complex contexts with many relations between the data objects (Maier et al., 2003).

So far, the CIB has been tested against small to medium size data quantities and has been able to fulfill all requirements. However, in future tests the complexity and amounts of real plant data will prove the applicability for real world data in the chemical industry. Therefore, the aforementioned ability to combine conventional database and semantic technologies in the CIB will be exploited.

4 CONCLUSIONS

This contribution reports on a current research project at the authors’ institute that deals with the development of a prototypical software tool for the integration and reconciliation of distributed design data like they are arising in a typical design project in chemical engineering. Based on the existing

integration solution PlantXML, which provides syntactic and structural homogeneous data sets accomplished by means of XML, the tool particularly aims at resolving semantic heterogeneities between the distributed information by defining an explicit representation of a data source's semantics by means of ontologies. To that end, the prototype incorporates the formal ontology OntoCAPE for the representation of the particular domain knowledge. Ultimately, the tool will extract, merge, and consolidate data from files in the PlantXML format in order to create a comprehensive information base (CIB). As a result, the further processing in the CIB will provide a detection and visualization of design errors.

The CIB, executing the information integration and reconciliation, is implemented in the design environment OntoStudio. The OntoCAPE ontology, is represented in the deductive ontology language F-Logic within OntoStudio.

REFERENCES

Anhäuser, F., Richert, H., Temmen, H., 2004. PlantXML-integrierter Planungsprozess mit flexiblen Bausteinen. *atp- Automatisierungstechnische Praxis* 46:10, 61-71.

Brandt, S.C., Morbach, J., Miatidis, M., Theißen, M., Jarke, M., Marquardt, W., 2008. An ontology-based approach to knowledge management in design processes. *Comp. Chem. Eng.*, 32, 320-342.

Yang, A., Braunschweig, B., Fraga, E., Guessoum, Z., Marquardt, W., Nadjemi, O., Paen, D., Piñol, D., Roux, P., Sama, S., Serra, M., Stalker, I., 2007. A multi-agent system to facilitate component-based process modelling and design. *Comput. Chem. Eng.*, in press.

Cruz, I., Xiao H., Hsu F., July 7-9 2004. An Ontology-based Framework for XML Semantic Integration. *Eighth International Database Engineering and Applications Symposium*, IDEAS'04, Coimbra, Portugal, University of Illinois at Chicago.

Embury, S.M., Brandt, S.M., Robinson, J.S., Sutherland, I., Bisby, F.A., Gray, W.A., Jones, A.C., White, R.J., 2001. Adapting integrity enforcement techniques for data reconciliation. *Inf. Syst.*, 26(8), 657-689.

Erdmann, M., Studer, R., 2001. How to structure and access, XML documents with ontologies. *Data & Knowledge Engineering*, 36(3), 317 – 335.

Fedai, M., Drath, R., 2004. CAEX – ein neutrales Austauschformat für Anlagendaten – Teil 1. In: *atp-Automatisierungstechnische Praxis* 46:2, 52-56.

Galaher, M.P., O'Connor, A.C., Dettbarn Jr., J.L., Gilday, L.T., 2004. Cost analysis of inadequate interoperability in the U.S. capital facilities industry. In: *Technial Report NIST GCR* 04-867. NIST-National Institut of Standatds and Technology.

Kifer, M., Lausen, G., Wu, J.. Logical foundations of object-oriented and frame-based languages. In: *Journal of the ACM* (42:4).

Klein, M., 2002. Interpreting XML Documents via an RDF Schema Ontology. In: *13th International Workshop on Database and Expert Systems Applications* (DEXA'02), dexa, 889.

Lenzerini, M., 2002. Data integration: A theoretical perspective. In: *Proc. Of the 21st ACM SIGACT SIGMOD SIBART Symp. On Principles of Database Systems (PODS 2002)*, 233-246.

Levy, A. Y., 2001. Answering queries using views: a survey. *VDBL Journal*

Maier, A., Aguado, J., Bernaras, A., Laresgoiti, I., Pedinaci, C., Pena, N., Smithers, T., 2003. Integration with ontologies. In: Reimer, U., Abecker, A., Staab, S., Stumme, G., (eds.): *Professionelles Wissensmanagement – Erfahrungen und Visionen*, Beiträge der 2. Konferenz Professionelles Wissensmanagement.

Morbach, J., Marquardt, W., 2008. Ontology-Based Integration and Management of Distributed Design Data. To appear in: Nagl, M., Marquardt, W., (eds.): *Collaborative and Distributed Chemical Engineering Design Processes: From Understanding to Substantial Support*, Springer. Berlin, Chapter 7.1.

Morbach, J., Theißen, M., Marquardt, W., 2008. Integrated Application Domain Models for Chemical Engineering. To appear in: Nagl, M., Marquardt, W., (eds.): *Collaborative and Distributed Chemical Engineering: From Understanding to Substantial Design Process Support*, Springer. Berlin, Chapter 2.6.

Morbach, J., Yang, A., Marquardt, W., 2007. OntoCAPE – a large-scale ontology for chemical process engineering. *Eng. Appl. Artif. Intel.*, 20(2), 147-161

Noumonon Consulting Limited, 2006. Open Access to Intelligent Process Plant models. Website: http://www.noumenon.co.uk/XMpLantOverview.html.

OntoCAPE, 2007, Access OntoCAPE, Website: http://www.avt.rwth-aachen.de/AVT/index.php?id=486

OntoStudio, 2007, OntoStudio Website: http://www.ontoprise.de/content/e1171/e1249/index_eng.html

Stuckenschmidt, H., Klein, M., 2003. Integrity and Change in Modular Ontologies. In: *Proceedings of the International Joint Conference on Artificial Intelligence - IJCAI '03*, Acapulco, Mexico, Morgan Kaufmann, 900-905.

Uschold, M., Gruninger, M., 1996. Ontologies: Principles, Methods and Applications. In: *Knowl. Eng. Rev.*, 11, 93-155.

Yang, A., Marquardt, W., 2004. An Ontology-Based Approach to Conceptual Process Modeling. In: Barbarosa-Póvoa, A., Matos, H., (eds.): *Proceedings of the European Symposium on Computer Aided Process Engineering – ESCAPE 14*, Elsevier, 1159-1164.