CONCEPTUAL MODELS FOR METADATA INTEGRATION AND ARCHITECTURE EVOLUTION

Christian Lettner, Christian Hawel, Bernhard Freudenthaler
Software Competence Center Hagenberg GmbH (SCCH), Softwarepark 21, Hagenberg, Austria

Ewald Draxler
voestalpine Stahl GmbH, voestalpine Straße 3, Linz, Austria

Keywords: Architecture, Conceptual Model, Integration, Metadata, Ontology, Semantic Technology, Topic Maps.

Abstract: Typical organizations operate several systems to perform their business tasks. To support the decision making process, the data processed by these systems must be integrated into a global, consolidated view. As each system supports a different business process and therefore operates in a different context, the data stored in these systems can have slightly different semantics. This semantic gap is one of the main reasons that data integration is a difficult task. The same challenges also apply to technical and business metadata. To resolve this semantic gap, conceptual models are proposed as the foundation for metadata integration. They are used to identify, interconnect and evaluate the similarity of concepts and to provide a vital source for architectural analysis. The conceptual model should provide a sound understanding of the domain and act as a general entry point for users who want to learn more about the system architecture.

1 INTRODUCTION

In most organizations the information system architecture is grown over time. This is also true for the range of functionalities these systems provide, as well as the underlying data models. If new functionality has to be implemented, the quickest and cheapest strategy is to implement it within the existing system of the requesting user. Furthermore, implementing the new functionality from scratch would fulfill the purpose it was requested for. Looking at other systems and reusing the existing functionality seems to be a more exhaustive and stressful way of developing extensions. This is especially true when the remote functionality performs the task slightly different than is needed by the new request. All these conditions stimulate the emergence of non-integrated information systems that are very difficult to manage. The problems can become virulent when corporate reporting demands integration of information from various systems into a single version of the truth.

One problem with non-integrated information architectures is that a so-called metadata map across all systems is often missing. This can lead to a situation where nobody is able to grasp the complete data architecture (i.e., nobody has an overview of which data is processed in which systems). Because various systems process the same information in a different context, understanding the exact meaning of an information element in system A compared to system B gets even more challenging. To get a complete picture of an organization's data architecture this semantic gap must be expressed in the organizational metadata map. The metadata of the individual systems has to be enriched with semantic information that enables access to a global, integrated view of the organization.

Migrating from a non-integrated information system to a single version of the truth takes time. In most cases, the first step is to reach a consensual catalog of vocabulary, which is valid for the entire organization. This catalog, also called a controlled vocabulary, must reflect semantic differences between terms used in different organizational units. Once such a vocabulary is established, the next step is to connect the vocabulary to the metadata of the individual systems. As a result, the individual systems metadata gets integrated at the semantic

Lettner C., Hawel C., Freudenthaler B. and Draxler E..
CONCEPTUAL MODELS FOR METADATA INTEGRATION AND ARCHITECTURE EVOLUTION.
DOI: 10.5220/0003094802690275
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2010), pages 269-275
ISBN: 978-989-8425-29-4

level through the common vocabulary. This integrated metadata serves as the foundation for the integration at the instance level, to finally provide the requested single version of the truth.

In this work we focus on the first two steps: establishing a control vocabulary and using this vocabulary to perform metadata integration across individual systems. We propose to use conceptual models for these tasks. Conceptual models are well-known in the data engineering community, but they are mainly used for analysis and design phases. The conceptual model should act as a general entry point for end users; it should provide knowledge about individual systems to end users.

Chapter 2 discusses related work in the field of metadata integration, data integration and semantic technologies. Chapter 3 introduces conceptual models based on Topic Maps. Chapter 4 shows how conceptual models can be used to support the metadata management and architecture evolution process. Finally, conclusions and suggestions for further work are provided in chapter 5.

2 RELATED WORK

According to Gilliland (2008, p. 2), metadata is "the sum total of what one can say about any information object at any level of aggregation". It is data about data. Data integration efforts are all about integrating metadata from different sources. This is often difficult, especially if the sources are heterogeneous. Haslhofer and Klas (2010) classify predominant metadata heterogeneities mentioned in the literature. They distinguish between two classes, structural and semantic heterogeneities. Structural heterogeneities arise as facts are modeled in different ways and with different levels of detail. One example of this so-called meta-level discrepancy is if location information in one system is modeled as an attribute while in the other system it is modeled as an entity. Semantic heterogeneities occur because of different semantics in the models. An example of this class is a terminological mismatch, when two attributes or entities with different names represent the same concept. Haslhofer and Klas (2010) classify different techniques to overcome these heterogeneities, such as controlled vocabularies, global conceptual models, etc.

A method of implementing these techniques is semantic technologies. Semantic technology, more precisely semantic web technology, has been developed to provide a better architecture that allows easy information integration from multiple web sites. As Horrocks (2008) notes, using semantic technologies like ontologies for information integration is not a novel approach and has been the subject of extensive research in the database community.

As given in Horrocks (2008), an ontology introduces a vocabulary, it describes various aspects of the domain being modeled and provides an explicit specification of the intended meaning of that vocabulary (Gruber, 1993). Ontologies can play several roles in organization for data integration:

- they describe the structure and semantics of data sources,
- they provide a detailed model of the domain against which queries are formulated,
- they provide a controlled vocabulary,
- they are used to identify, associate and finally integrate semantically corresponding concepts (Horrocks, 2008) (Wache et al., 2001).

As depicted in Gilliland (2008) the complexity of a controlled vocabulary can range from a simple list of terms to a system that defines terms and the semantic relationships between them. Using ontologies, the later can be accomplished. However, ontologies go further than simply representing knowledge. They provide information about the domain of interest and derive new knowledge based on the facts modeled in the ontology. Because reasoning about a domain was not important in this work, ontologies were not used. Instead, Topic Maps (ISO/IEC, 2002), a simpler language built for representing knowledge, has been used.

Identifying, associating and integrating semantically corresponding concepts is a very similar problem to schema matching and data matching in the database community. In Doan and Halevy (2005), two groups of semantic matching techniques were distinguished, rule-based solutions and learning-based solutions. Rule-based solutions operate only on schemas, are very fast and do not require training. Learning-based solutions exploit data instances, which can hold very valuable information for schema matching. For this reason they are not as fast as rule-based solutions and require training. As already mentioned, data matching or tuple matching can be considered as a similar problem like schema matching. As expressed in Doan and Halevy (2005), research in both disciplines supports one another. Duplicate detection is considered to be one of the most prominent data matching techniques.

Traditional data integration is performed based

on precisely engineered mappings. A data element will be extracted from a source schema, will get transformed and finally will be loaded into a destination schema. Mappings from the source to the destination schema must be completely defined before data integration can take place.

Dataspace management systems try to perform data integration processes the other way round. Inspired by desktop search engines, a basic keyword query utility is available and can be used over all data sources to be integrated. Initially, the quality of integration is limited to providing a common keyword search interface. Like developing integration mappings, no prior investment in the integration process is needed. The result set will contain duplicates, different data encodings, etc. As the data integration requirements get more demanding, the dataspace management system must be enriched with more relationships and mappings. Using this additional information the integration process should be able to produce a better level of integration. This process is called pay-as-you-go (Salles, Dittrich, Karakashian, Girard and Blunschi, 2007) (Franklin, Halevy and Maier, 2005).

A system that allows for gradual enhancement of relationships is proposed in Talukdar et al. (2008). Starting with a basic keyword search, the system matches the keywords to relational data sources and evaluates available associations to finally generate a list of possible queries ranked by a score. Next, the user has to provide feedback to the system regarding which query fits the intended information request best. Based on this feedback, the system is able to learn by assigning new weightings to associations, which in turn changes the future rank of proposed queries.

In this work, the idea of gradually enhancing the level of integration is applied when mapping the controlled vocabulary to the logical data models of the systems in an iterative way. Initially, only a basic keyword search will be available over the controlled vocabulary and the logical data models of the systems. As more and more mappings are introduced, the more the semantic context of the system will be formed, which in turn closes the semantic gap across the systems.

A similar approach has been proposed in Karjikar, Roy and Padmanabhuni (2009), but for another area of application. It used Topic Maps to represent the knowledge contained in Universal Business Language (UBL) documents, an OASIS standard for generic business documents.

3 CONCEPTUAL MODELING USING TOPIC MAPS

Conceptual models provide a controlled vocabulary and describe entities and relationships involved in these applications. Several languages exist for conceptual modeling. The most prominent representative is UML, the Unified Modeling Language (Object Management Group, 2007).

To close the semantic gap across systems, a semantic-aware conceptual modeling language is needed. Ontology oriented languages like Topic Maps and OWL, the Web Ontology Language, provide built-in support for semantic descriptions and thus seem to be most suitable. Topic Maps have been selected as the conceptual modeling language for this study because of its simplicity and practicality for end users. Topic Maps (ISO/IEC, 2002) are an ISO standard for knowledge representation. They are inspired by semantic networks and index structures. “Ontopia”, a general purpose open source Topic Maps development environment has been chosen for prototype development (Ontopia, 2010).

In the following, only the most important concepts of Topic Maps can be introduced. A detailed introduction is provided in Pepper (2010). The main model elements of Topic Maps are topics, associations and occurrences. Every topic has a type. Types can be inherited (i.e., every type can define parent types and sub types). Associations express relationships between one or more topics. The following example establishes an association “is-manager-of” between the topics “Christian” and “Michael”. Both topics are of type “Employee”.

```
Christian:Employee is-manager-of
Michael:Employee.
```

Occurrences are references to additional information that is relevant for the topic. This could be a reference to a web-site or simply a data element that contains additional information, such as the employee’s birth date.

A prototype of the proposed approach has been implemented for voestalpine Stahl GmbH. Voestalpine operates a lot of autonomous information systems, each designed to perfectly perform a certain task in the production process. The data processed in these systems must be integrated into a global view to optimally control the production process. As already mentioned, at a specific size, this distributed architecture gets very uncomfortable. Even experts find it difficult to give answers to simple questions like: “*Tell me all systems where some data element gets processed*.”

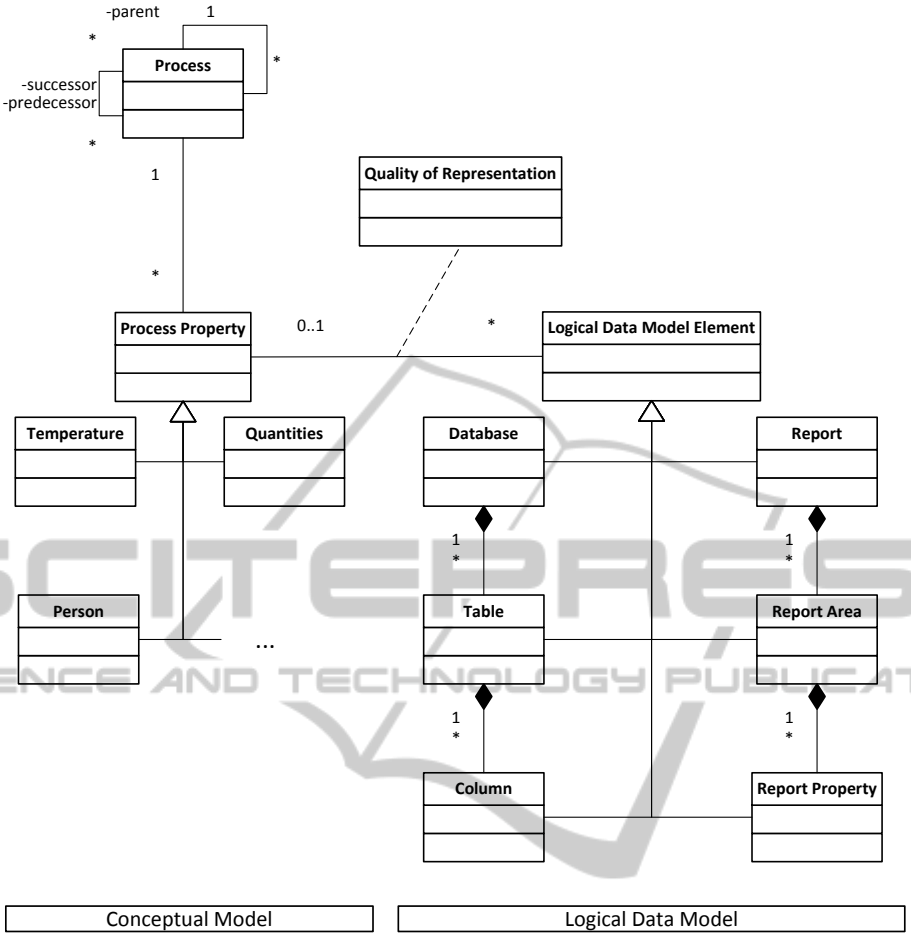

Figure 1: Class diagram of the domain and logical data model.

One aim of the prototype was to answer questions like these.

As production is mainly organized around processes, they also form the main topic in the conceptual model. Each process can be composed of multiple sub-processes and can have multiple predecessors and successors. Each process owns a list of process properties that characterizes the process. For reasons of readability "process properties" is substituted with "property" in the following. For each property a topic has been created that provides information about which data is available for the process. Examples for properties would be "quantities produced", "production start time" or "responsible employee". Each property has a type depending on its characteristics. Types include temperatures, quantities, etc., as well as properties from production facilities, such as pans. An example instance for a production facility property is "number of first pan used". These properties represent the different roles a pan can play in the complete production process.

Properties characterize the production process and have a representation in information systems (i.e., in logical data models of the information systems). Furthermore, representations can also exist on reports, production journals, etc. Some properties have semantically equivalent representations in more than one system or artifact.

Associations are modeled between the property in the conceptual model and the corresponding representation in the systems logical data model. If the property has representations in more than one system, then these associations constitute the clue that links semantically corresponding concepts across systems. As in dataspace management systems, these associations can be gradually enhanced to move the systems more closely together.

Depending on the system, the data quality of the representation can vary significantly across the systems. Examples of data quality attributes are the number of missing or duplicate values. To capture knowledge about data quality, quality of

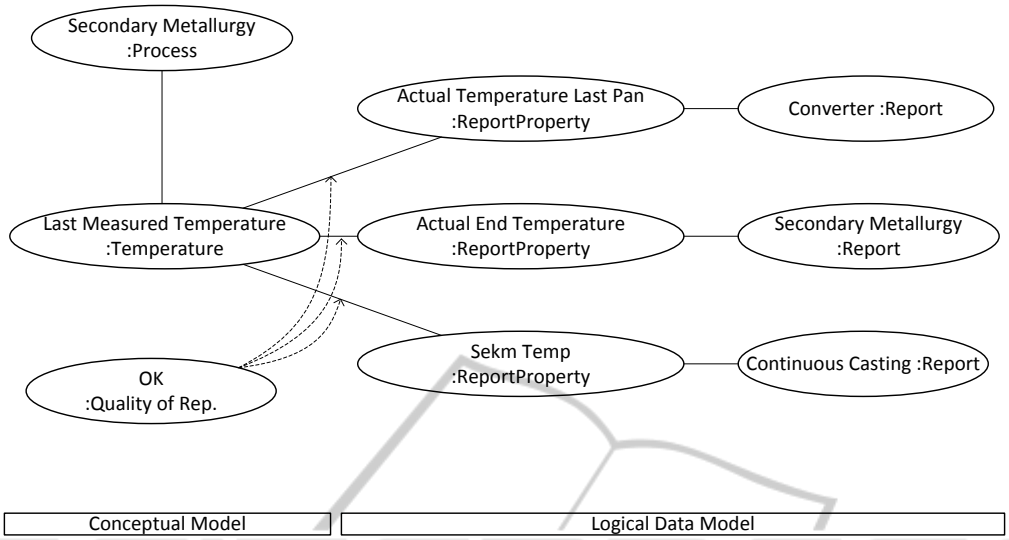

Figure 3: Instance diagram of conceptual model and logical data model.

representation can be specified for the associations between the conceptual model and the logical data model. Figure 1 shows a class diagram of the conceptual model used and the association to the logical data models of the systems. Both the conceptual model and the logical data models are modeled as Topic Maps.

Figure 2 shows excerpts from three different reports, each displaying the same temperature, marked with bold squares. The problem is that each report uses another label for the temperature, respectively "Temperatur Ist Letzte Pfanne" ("Actual Temperature Last Pan"), "Temp Ist End" ("Actual End Temperature") and "Sekm Temp"

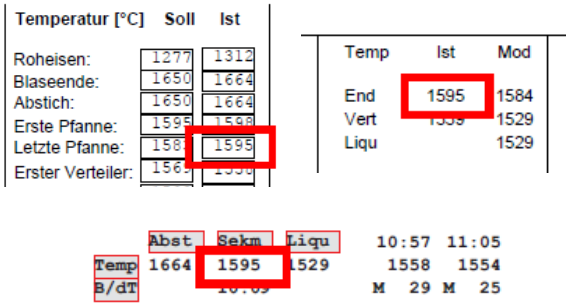

Figure 2: Three reports showing the same temperature.

("Secondary Metallurgy Temperature"). Only domain experts know that all three temperatures actually mean the same thing. For non-experts, this knowledge remains hidden.

An instance diagram that demonstrates how this situation is modeled in the conceptual model with the logical data model is presented in Figure 3. The conceptual model is used to capture the semantic equality of the three temperatures displayed in the reports. There only exists one temperature process property, called "Last Measured Temperature", in the domain model, which belongs to the process "Secondary Metallurgy". Starting from this process property there are three associations to the corresponding report properties in the reports. All three representations have been classified as "OK" as the quality of representation.

4 USING THE CONCEPTUAL MODEL

To provide the modeled knowledge in the conceptual model to a major group of users, easy ways to access the information must be available. In this section, an overview of the possibilities to access the conceptual models is given.

4.1 Keyword Search

The easiest way for users to access information is to use a keyword search. As the Topic Maps tool "Ontopia" provides a basic keyword search utility, keyword search is available for the conceptual model and logical data models. As a result, the name and the type of the found topics are displayed. The search not only covers the name of the topic, but also the type information available in the conceptual model. Figure 4 illustrates an example for the keyword search. Searching for the word

"Temperature" as well. "temperature" delivers all topics of type

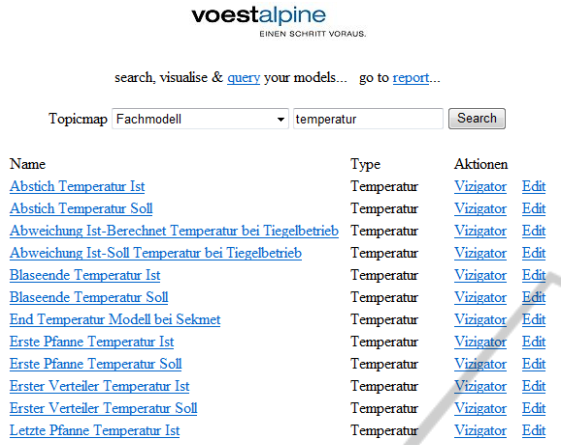

Figure 4: Keyword search on the conceptual model.

4.2 Navigate by Associations

Another way to approach the information modeled in the conceptual model is using associations. For example starting from a well-known process, the association structure to parent-, sub-, previous- and successor-processes can be used to navigate to the information the user is interested in. The navigation can be assisted using the graphical association visualization tool "Vizigator" that ships with "Ontopia".

Figure 5 shows such an example for a process navigation application. "Schmelzmetallurgie" (melting shop), "Brammenfertigung" (slab caster) and "WB-Fertigung" (hot rolling mill) illustrate one segment of the workflow of the described domain. The process „Schmelzmetallurgie" has two sub-processes, "Primäre Metallurgie" (primary metallurgy) and "Sekundäre Metallurgie" (secondary metallurgy) which are connected by using associations. Each of these sub-processes can consist of other sub-processes. For example, the sub-process "Primäre Metallurgie" has sub-processes like "Tiegelpflege" (converter maintenance) or "Schlacke kippen" (slag tipping) which can be associated again. "Roheisen" (pig iron) is an example for materials and demonstrates the material flow to "Schmelzmetallurgie".

Another approach is to build a custom application that uses association information stored in Topic Maps to provide a domain specific presentation of the stored information. Based on the association type, the custom application arranges the topics accordingly, to allow intuitive navigation. Figure 6 shows such an example for a process

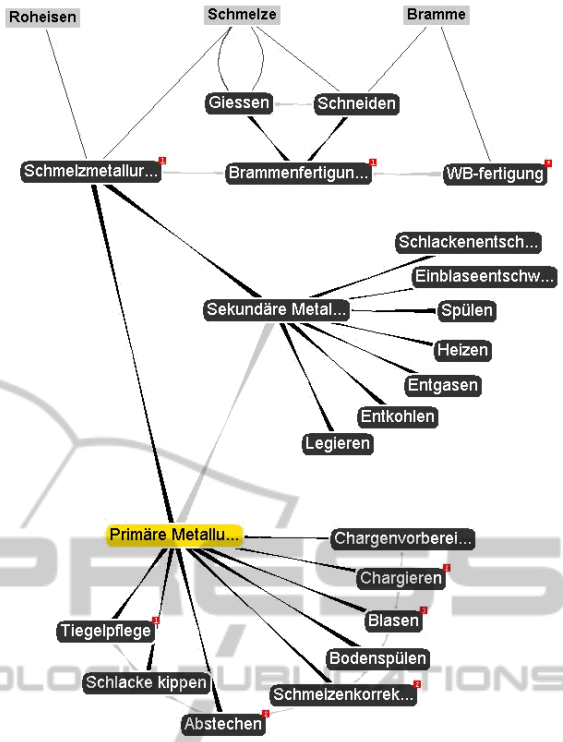

Figure 5: Using "Vizigator" for process navigation.

Figure 6: Custom process navigation application example.

navigation application. The current selected process is "Blasen". The super-process of "Blasen" is "Primäre Metallurgie" while the previous process is "Chargieren" and the following process is "Bodenspülen". "Blasen" also has three sup-processes, called "Sublanzenphase Blaseende", "Blasebeginn/Zünden" and "Hauptblasen". The

process properties for "Blasen" are presented in the lower part of the screen.

4.3 Query the Conceptual Model

"Ontopia" provides a Topic Maps query language called "tolog". "tolog" is a Datalog like query language for Topic Maps. It can be used to perform architectural analysis on the conceptual model and logical data model. For example, queries like "*Give me all temperatures that are processed in System A*" can be written easily. Further, queries can be formulated that allow monitoring the system architecture. For example, if a new entity will be added to the logical data model that has no mapping in the domain model, a reminder or message will be generated to add this missing information. The possibility to monitor the system architecture is a key enabler for a vital conceptual model and system architecture.

5 CONCLUSIONS AND FURTHER WORK

In this work, Topic Map based conceptual models are proposed for metadata integration. A conceptual model is used to provide a lasting and intuitive source of information on how the metadata of the individual systems is semantically related to each other. Currently, conceptual models and their relationships to logical data models must be largely defined manually. Actuality of the conceptual model and its relationships to the systems is crucial for acceptance by users. As such, further work will concentrate on developing semi-automatic methods to maintain the actuality of the models.

Further, we will consider using ontologies instead of Topic Maps as they are much more expressive. For example, one shortcoming of Topic Maps is the absence of a practical concept to manage synonyms, which is essential when establishing a controlled vocabulary.

Other possible future work could concentrate on introducing different levels of abstractions for conceptual models. Different users need different levels of conceptual model detail for their work. A language built-in concept that allows composing higher level topics out of a group of basic topics would be preferable. The user could decide for herself/himself, weather s/he wants to see a high level view of the domain or browse into the details of the domain.

Finally, a future direction could be to concentrate on how the conceptual model can be used to perform data integration at the instance level, also called ontology-based data access. Ontology-based data access uses queries specified on a conceptual model to access and integrate data from underlying systems.

REFERENCES

Doan, A. and Halevy, A. (2005). Semantic-Integration Research in the Database Community. AI Magazine, Vol. 26, No. 1.

Franklin, M.; Halevy, A. and Maier, D. (2005). From Databases to Dataspaces: A New Abstraction for Information Management. ACM SIGMOD Record, Vol. 24, No. 4.

Gilliland, A. (2008). Setting the Stage. In Baca, M. (Ed.), Introduction to Metadata (2nd ed.). Getty Publications.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition. Vol. 5, No. 2, p. 199-220. London: Academic Press Ltd.

Haslhofer, B. and Klas, W. (2010). A survey of techniques for achieving metadata interoperability. ACM Computing Surveys, Vol. 42, No. 2, Article 7.

Horrocks, I. (2008). Ontologies and the Semantic Web. Communications of the ACM, Vol. 51, No. 12.

ISO/IEC. (2002). ISO/IEC 13250 Topic Maps (2nd ed.).

Karjikar, F.; Roy, S. and Padmanabhuni, S. (2009). Intelligent Business Knowledge Management Using Topic Maps. Compute'09: Proceedings of the 2nd Bangalore Annual Compute Conference, p. 1-8.

Object Management Group. (2007). Unified Modeling Language (UML). Retrieved from: http://www.uml.org.

Ontopia. (2010). Ontopia. Retrieved from: http://code.google.com/p/ontopia/

Pepper, S. (2010). The TAO of Topic Maps. Oslo. Retrieved from: http://www.ontopia.net/topicmaps/materials/tao.html.

Salles, M. A. V.; Dittrich, J.; Karakashian, S. K.; Girard, O. R. and Blunschi, L. (2007). iTrails: Pay-as-you-go Information Integration in Dataspaces. VLDB'07: Proceedings of the 33rd international conference on Very large data bases, p. 663-674.

Talukdar, P.P.; Jacob, M.; Mehmood, M.S.; Crammer, K.; Ives, Z.G.; Pereira, F. and Guha, S. (2008). Learning to Create Data-Integrating Queries. VLDB'08: Proceedings of the VLDB Endowment, p. 785-796

Wache, H.; Vögele, T.; Visser, U.; Stuckenschmidt, H.; Schuster, G.; Neumann, H. and Hübner, S. (2001). Ontology-Based Integration of Information – A Survey of Existing Approaches. IJCAI-01 Workshop: Ontologies and Information Sharing, p. 108-117.