Using Semantic Technologies for More Intelligent Steel Manufacturing

Nikolaos Matskanis[1], Stephane Mouton[1], Alexander Ebel[2] and Francesca Marchiori[3]
[1]*Centre of Excellence in Information & Communication Technologies (CETIC), Rue Freres Wright 29, Gosselies, Belgium*
[2]*VDEh-Betriebsforschungsinstitut (BFI), Düsseldorf, Germany*
[3]*Centro Sviluppo Materiali (CSM), Rome, Italy*

Keywords: Data Integration, Data Interoperability, Domain Ontology, Semantic Services, Semantic Software Agents.

Abstract: In recent years, the steel industry has significantly raised its demands regarding product quality, optimization of production cost, environmental issues and lead-time. The demand for improved production performance has in turn increased the demand on information systems, in particular highlighting the need for improved factory- and company-wide collaboration and information exchange. The heterogeneity in structure, technology and architecture of the information systems deployed in manufacturing plants presents further challenges to the design and implementation of a data exchange system for process optimization.

1 INTRODUCTION

This article proposes a system that can be used to increase collaboration of heterogeneous information systems as part of a platform that aims to optimize production and product delivery of steel manufacturing plants. The proposed architecture uses a distributed paradigm that: i) enables simultaneous and easy exchange of mutual data; ii) provides interoperability in the data structure, data semantics and message exchanging; and iii) applies rules and data reasoning techniques to optimize resource allocation.

This service oriented, semantic interoperability solution is being implemented in the context of an systems integration project in the Steel Manufacturing domain. It is a project funded by an EC RFCS program and its consortium consists of both large industrial partners and small or medium enterprises focusing in technological research and development. The project represents a new approach to controlling and supervising steel manufacturing processes that uses Agent technology (Haag and Cummings, 2006) to infuse intelligence into the control system. Each agent is a software component that is characterized by a large degree of autonomy, and the resulting multi agent system will be capable of integrating semantics and will be easily deployable in distributed environments. One use case that we are considering for the semantic interoperability services is product re-allocation: i.e., steel products that are either not compliant with their initial order or are over-produced are re-allocated to a new order. The re-allocation may include further product processing at any of the plants. Additionally takes into consideration transportation costs, costs of trimming or other processing of the product in order to comply with the order specifications.

2 SEMANTIC INTEROPERABILITY SERVICES

One of the platform's main challenges is to enable software components, which are agnostic of the local and remote information systems, to request data from data stores and sensors and seamlessly perform optimization operations despite the syntactic/structural and semantic heterogeneity of the different data models of steel plants. The structural heterogeneity issues are related to the way information is represented at each data source. The semantic heterogeneity relates to the use of different terms, languages for referring to the same concept or different definitions of the same entity.

Matskanis, N., Mouton, S., Ebel, A. and Marchiori, F..
Using Semantic Technologies for More Intelligent Steel Manufacturing.
In *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015) - Volume 2: KEOD*, pages 424-428
ISBN: 978-989-758-158-8

2.1 Related Work

The idea of using Web Services and Semantic Web technologies for establishing data integration and interoperability is an area of scientific research that is being explored in several application domains and several approaches already exist and some solutions have been developed. In (Yang et al., 2005) paper it is described a Web Service Oriented Architecture (SOA) and platform that uses semantics and specifically OWL-S for automatic integration of manufacturing systems. It focuses on the dynamic discovery, selection of web services and other business processes that can be described using OWL-S. Another approach is presented in (Uddin et al., 2011), which focuses on the ontology and formal description of data sources using a domain ontology. Finally the (Chondrogiannis et al., 2011) and (Martin et al., 2008) approaches are accomplishing interoperability of heterogeneous medical information systems by transforming SPARQL queries to the different clinical ontologies and code systems that they have linked as well as different database schemas that are mapped to domain ontologies. In their approaches the linking between the ontologies is done manually or the transformation is performed by using term transformation services.

In our approach we have designed and built the domain ontology and the plant specific extensions with concept inter-linking in in mind. This linking between the concepts is encoded in the ontologies either with the form of inheritance or by using semantic relationship links between them. By having the concepts of the ontologies inter-linked, building services that offer query and schema transformation and mapping across different manufacturing plants becomes easier since the mapping information is already encoded in the ontologies. This enables software agents to be domain agnostic and query the plant data using the platform's upper terminology.

2.2 Service Architecture

In order to achieve the goal of enabling the platform's software components to automatically resolve both heterogeneity types, we have introduced into the architecture and deployed on the platform semantic interoperability services. These services use ontologies (Berners-Lee et al., 2001) that describe the steel-manufacturing domain and provide a formal definition of the types, properties and interrelationships of the entities in this domain model. The ontologies are structured into two layers, the high-level domain ontology and the lower level plant ontologies.

Interoperability is achieved by linking the concepts/terms of the high-level (core) steel domain ontology, which is being developed in the project (Zillner et al., 2014) to provide a common terminology for all participating plants, and the plant specific ontology, which provide the terminology, description of methods and model used in each plant. These links are implemented either by an inheritance relation between entities or by explicitly defined interrelationships between entities of the high–level and plant ontologies, for example the Web Ontology Language (OWL) property owl:sameAs. We are also developing a 3rd level which maps the database schema with the plant ontology and effectively adds a SPARQL (Query Language for RDF) interface for the database. This way the plant databases can also be accessed and exposed as RDF (Resource Description Framework) graphs and effectively queried the same way as ontologies. The proposed architecture design and proof of concept implementation of the semantic service using the domain ontology, plant ontologies and plant data sources is shown in Figure 1.

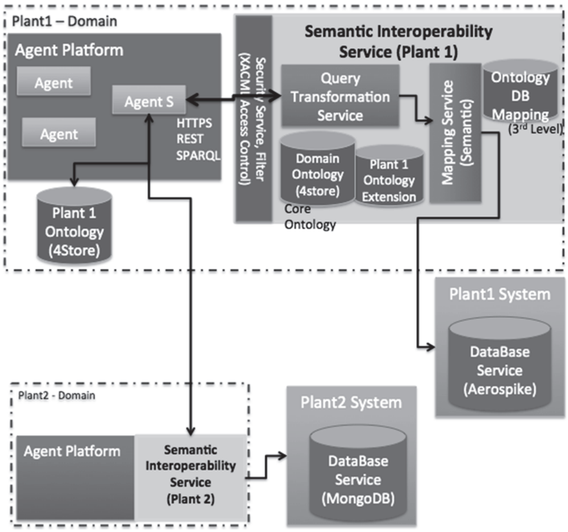

Figure 1: Architecture of the proof of concept platform. The semantic and other services are presented.

The semantic mapping between the ontology layers creates associations of related concepts of the domain ontology, the plant ontology and the actual data in the data stores. This approach allows translation of the client query from the domain ontology that all - participating in the project platform - plants are able to understand and use, to the local one used in each of the plants, and eventually – if requested – to the database native query language. This approach is generic enough to be reused in other applications, for example a similar solution was applied in the clinical research domain (Chondrogiannis et al., 2011). The reallocation use case of the project, which involves multiple plants with an agent platform each, benefits from this approach when agents of different plants i) request for product data values, ii) query a plant's model to identify manufacturing processes, logistical or other plant information.

2.3 Proof of Concept Implementation

2.3.1 Technologies and Tools

Several tools and technologies exist that can be used for the implementation of a semantically enabled service for data integration and interoperability. For the development of the ontology we have considered the Protégé ontology editor but as our requirements were for an easily accessible and demonstrable environment for the plant engineers, the Semantic Media Wiki (SMW) (Vrandečić and Krötzsch, 2009) appeared to be more appropriate for this purpose. The backend of the Media Wiki is a MySQL database server and for storing the OWL RDF triples of the SMW extension we have used the 4store triplestore. 4store provides a SPARQL endpoint and is used for storing and querying both the Domain Ontology and the plant specific ones. The choice of the triplestore was based on its compatibility with the SMW and because the set of requirements for the triplestore technology was rather small, a lightweight server was preferred from the feature-rich OpenLink Virtuoso server that we were also considering.

Another key part of the architecture is the mapping of the plant ontology and the data itself. Several approaches were investigated. From those we have tested the Virtuoso server, the Semantika server (Hardi, 2014) and the D2RQ server (Bizer and Seaborne, 2004). An additional complication and challenge was the nature of the Databases we have used in the project for the plant products and customer orders. These two datasets are stored in a NoSQL document-oriented database (Sadalage and Fowler, 2012), the MongoDB, which dynamically adjusts its schema of the database to the data that is stored in each document, which in our case it is each entry in the database. The absence of concrete schema introduced problems to the mapping server of the ontology with the database, which was solved using the unityJDBC (www.unityjdbc.com) driver with the D2RQ server. The D2RQ server does not require validating the schema, unlike Semantika, and the unityJDBC driver provides the core ODBC functionality and API. This enabled the plant product and orders databases to be accessed as RDF Triplestores using a SPARQL endpoint and be browse-able by both humans and machines as Linked Data (Heath et al., 2010), though we have not openly published them.

Finally the Semantic agents implement the recommendation of FIPA Agent Management Specification, which suggests integrating the ontologies by providing so called "Ontology Services". In this approach the access to the external ontology server by internal agents is organised through a specialised agent who is providing the ontology services. These services include forming the SPARQL queries, utilising the semantic links and mappings and of course accessing the semantic services described above. The internal agents can use the same communication mechanism to access the ontology as for the intercommunication between them. The semantic agent can implement additional functionalities such as transformation of data returned by the ontology or provide knowledge about mappings between the different ontologies. Figure 2 shows a general scheme of the semantic agent interaction with the ontology services. Here the sematic agent provides the functionality to provide a term mapping.

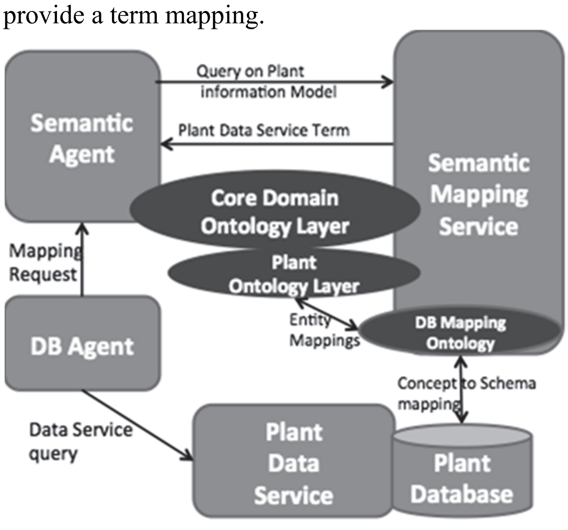

Figure 2: Retrieving local term based on the domain ontology.

2.3.2 Data Querying

The first use case for the semantic services is to directly retrieve data from the databases using SPARQL queries. The mapping of the D2RQ server between the plant ontology and the database schema allows translation of the SPARQL queries to SQL and retrieval of the requested data from the database. The semantic agent can either include the terms from both the domain and plant ontologies in the query or translate the query by replacing the domain terms with the plant ones. In our experiments so far we have used template queries that use the later approach.

2.3.3 Schema Translation and Mapping

In this use case of the semantic service, the semantic agent queries the ontology service for a term or concept that is used in the plant data schema. In order to produce the result the service requires a term translation from the core domain ontology to a term used in the database schema of the plant. The translation is achieved using the links between the ontologies that are structured into two layers, the high-level domain ontology and the lower level plant ones. The semantic links between concepts that were previously explained associate the related concepts of the two ontologies and with the addition of the URI of the database field provided by the D2RQ mapping, the complete path to the correct document and field of the database is produced.

This path is then cached to a JSON file in order to avoid having the semantic agent calling the ontology service every time there is a need to access the database. With this caching of mappings the agents of the platform can be agnostic of the schema of the plant database and only need to the domain ontology terms for the data they need to access. The format of the cached mapping is provided in the JSON code bellow:

```
[{"definition":"http://steel.eu/product.coil
",
"label":"Coil", ... ,
"dataAttributes":[{
    "definition":"http://steel.eu/product.co
il_width",
    "label":"Width", ...}],
"equivalentTo":{
"definition":"http://steelcompany.com/Coil",
"label":"Coil", ...,
"dataAttributes":[{
    "definition":"http://steelcompany.com/Wi
dth",
    "label":"Width", ...}],
"equivalentTo":{
    "definition":"https://steelcompany.com
/d2rq/resource/PRODUCTS/Sagunto/PRODUCTS_Sag
unto", ...,
    "dataAttributes":[{
    "definition":"https://steelcompany.com/d
2rq/resource/PRODUCTS/Sagunto/Width",
    "label":"Width", ...}]}
...}]
```

3 CONCLUSIONS AND FUTURE WORK

In this paper we describe the semantic services architecture and prototype that we have produced around the domain ontology of steel manufacturing that we have developed and its plant ontology extensions. The services use the semantic links between the ontologies and provide mappings to the data sources of the plants. The semantic agent connects the puzzle by creating the SPARQL queries based on templates.

The prototype that was developed was deployed in one of the partner plants in order to be part of the reallocation use case experiment of the project. Although the reallocation agent responsible for performing allocations of products had no information on the Products, Orders and other plant databases and their schemata, it had been successful in querying these data sources by using the project's core domain ontology and by performing schema term translation operations, which are being cached for performance reasons. This allows the project to extend its experiments by easily adding new plants that are distributed in many locations in Europe and although belong to the same organisation they have very different infrastructure and systems. One of the goals of the project is to allow dynamic adding or removing plants into the product reallocation service at runtime and we believe that this architecture and proof of concept implementation is a step towards this goal.

With the ontology mapping and semantic services in place we can further explore the reasoning capabilities on the ontology and data for deriving interrelationships, thus expanding the mapping between ontology and data, which can lead to identifying new product reallocations using the order and product data stores. This will further empower the resource allocation optimization processes of the platform.

REFERENCES

Berners-Lee, T., Hendler, J., Lassila, O., 2001, *The Semantic Web*, Scientific American, 284(5):34-43.

Zillner, S., et al., 2014, *A Semantic Modeling Approach for the Steel Production Domain*, 1th European Steel Technology & Application Days & 31th Journées Sidérurgiques Internationales (JSI), Paris.

Chondrogiannis, E., Matskanis, N., et al., 2011, *Enabling semantic interlinking of medical data sources and EHRs for clinical research purposes*, eChallenges conference.

Haag, S., Cummings, M., 2006, *Management Information Systems for the Information Age*, pp. 224–228.

Uddin, M. K., et al., 2011, *An ontology-based semantic foundation for flexible manufacturing systems*, 37th Annual Conference of the IEEE Industrial Electronics Society, IECON 2011, pp. 340–345.

Yang, Z., et al, 2005, *Automating integration of manufacturing systems and services: a semantic Web services approach*, 31st Annual Conference of IEEE Industrial Electronics Society, IECON 2005.

Martin, L., Tsiknakis, M., et al. 2008, *Ontology based integration of distributed and heterogeneous data sources in ACGT*, 1st International Conference on Health Informatics, HEALTHINF 2008, (vol. 1, pp. 301-306).

Bizer, C., Seaborne, A. 2004, *D2RQ – Treating Non-RDF Databases as Virtual RDF Graphs*, International Semantic Web Conference (ISWC2004).

Hardi, J., 2014, *Introduction to Semantika DRM,* http://www.codeproject.com/Articles/787371/Introduction-to-Semantika-DRM.

Vrandečić, D., Krötzsch, M., 2009, *Semantic MediaWiki. In John D. et al., eds.:* Semantic Knowledge Management. Springer.

Heath, T. Bizer C., Berners-Lee, T., 2010*. Linked data-the story so far*, International Journal on Semantic Web and Information Systems, Vol. 5(3), Pages 1-22. DOI: 10.4018/jswis.2009081901

Sadalage, P., Fowler, M., 2012*, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence,* Addison-Wesley, ISBN 0-321-82662-0.