General Adaptation Framework: Enabling Interoperability for Industrial Web Resources

Olena Kaykova, Oleksiy Khriyenko, Dmytro Kovtun, Anton Naumenko, Vagan Terziyan, Andriy Zharko

Industrial Ontologies Group, Agora Center, University of Jyväskylä, Finland
e-mail: vagan@it.jyu.fi

Abstract

Integration of heterogeneous applications and data sources into an interoperable system is one of the most relevant challenges for many knowledge-based corporations nowadays. Development of a global environment, which would support knowledge transfer from human experts to automated Web-services which are able to learn, is a very profit-promising and challenging task. The domain of industrial maintenance is not an exception. This paper outlines in details an approach for adaptation of heterogeneous Web resources into a unified environment as a first step towards interoperability of smart industrial resources, where distributed human experts and learning Web-services are utilized by various devices for self-monitoring and self-diagnostics. The proposed General Adaptation Framework utilizes a potential of the Semantic Web technology and primarily focuses on the aspect of a semantic adaptation (or mediation) of existing widely used models of data representation to RDF-based semantically rich format. To perform the semantic adaptation of industrial resources the approach of two-stage transformation (syntactical and semantic) is elaborated and implemented for monitoring of a concrete industrial device with underlying XML-based data representation model as a use case.

Keywords

Semantic Web, Industrial Maintenance, RDF, Interoperability, Data Transformation, Semantic Mediation, Adaptation Framework

1 Introduction

At the current stage of ICT development, there is a diversity of heterogeneous systems, applications, standards of data representation and ways of interaction. All those systems were tailored for particular tasks and goals. The world is heterogeneous and modern industry is looking for fast and global solutions related to Knowledge Management, Enterprise Application Integration, Electronic Commerce, Asset Management, etc. However, in spite of advancements in data processing and acquisition it is still difficult to automatically process and exchange data between the heterogeneous systems. Various industrial standards, which have been created and implemented by different consortia, appear to be not sufficient for growing interoperability demands.

Taking into account great variety of possible types of information resources, data formats and ways of data accessing and acquisition, an integration of such resources into a unified environment is an important development challenge [3, 14].

Basically, the integration tasks can be solved by adaptation of data from heterogeneous formats to some commonly accepted and semantically reach format, i.e. adaptation of heterogeneous applications and data originally represented according to different standard to common standard.

The integration process may include the following key functions [2, 32]:

- Extracting, transformation and loading – for building data warehouse or operation data stores and giving to an end-user/application a possibility to work with integrated data;
- Data replication, to allow heterogeneous servers and databases to share data in real time;
- Data Synchronization – to allow sharing of data between servers and remote devices when connectivity is temporary.

Application adaptation is a special part of the general integration task. The data is generated by different applications with specific features:

- Application functions;
- Application APIs;
- Application interfaces.

All variations of these features have effect on process of adaptation and architecture of adaptation framework.

During last several years, the major efforts in solving the challenge of Enterprise Application Integration have been focused on the domain of Web Services – loosely coupled internet and intranet applications developed according to the requirements of W3C's Web Services Architecture Working Group. So far, standardization efforts of W3C in this direction have resulted into SOAP [30], WSDL [40] and UDDI [39] specifications. Industry currently supports these standards as a solid solution for a wide variety of tasks from the EAI domain. Service-oriented approach is actively used in modeling Business-to-Business tasks and Business Process Execution Language for Web Services (BPEL4WS [4]) has been the most widely used nowadays.

Semantic Web is relatively new initiative within W3C standardization effort to enable machine interpretable metadata in the Web. It provides standards and tools to enable explicit semantics of various Web resources based on semantic annotations and ontologies. Integration in general is considered nowadays as a "killer application" of Semantic Web technology, which particularly can be interpreted as heterogeneous data integration, Enterprise Application Integration and Web-service integration among other interpretations. In contrast to ICT, the semantic technologies represent meanings separately from data, content, or program code, using the open standards for the Semantic Web. They are language neutral, machine interpretable, sharable, and adaptive, allow ontology based integration of heterogeneous resources. Automated knowledge accumulation and sharing is becoming the most profitable kind of business for modern knowledge-driven enterprises.

Academic community has been actively utilized concepts of Semantic Web for further development of the potential of Service-oriented analysis and underlying standards. Extending current XML-based standards for Web Services by explicit semantics would make automated discovery or composition of Web services possible. The recent efforts in the domain of Semantic Web Services are represented by three major projects: OWL-S [26], METEOR-S [20, 31] and WSMO [41], and associated initiatives such as SWWS [28], SWSI/SWSA [29] and interoperability initiatives between some of these [17, 27]. WSMO and several related activities are being performed within the European Adaptive Services Grid (ASG [1]) project. CASCOM European project [6] can also be mentioned as one of the significant project of the concerned domain, which is based on inter-disciplinary combination of intelligent agent, Semantic Web, Peer-to-Peer, and mobile computing technology.

One of the domains, where knowledge accumulation and its timely delivery are crucial, is industrial maintenance [21]. Development of a global environment, which would support

automation of knowledge management for industrial maintenance, is a very profit-promising and challenging task. The latter is what the Smart Resource[1] project aims at in the research and development efforts of Industrial Ontologies Group[2].

The intention of the SmartResource team is to provide tools and solutions to make heterogeneous industrial resources (files, documents, services, devices, processes, systems, human experts, etc.) web-accessible, proactive and cooperative in a sense that they will be able to analyze their state independently from other systems or to order such analysis from remote experts or Web-services to be aware of own condition and to plan behavior towards effective and predictive maintenance.

This paper presents an approach and a case study performed by SmartResource team within an industrial maintenance domain aimed to design a possible architecture for interoperability of heterogeneous industrial resources based on Semantic Web standards. Emphasis is made on a General Adaptation Framework, which is envisioned to enable reusable solutions and components for automatic adaptation of different types of resources and their data formats to Semantic Web environment.

The structure of the further content is the following. Section 2 introduces a Global Understanding Environment as a background concept for future implementation and also describes briefly stages of architectural design for it. Section 3 provides more design and implementation details about the Adaptation Stage for the target environment design. In Section 4 the general approach to building semantic adapters for industrial resources is given. The section describes the challenge itself, goes deeper into the details of the two-stage transformations between data models and presents its pilot implementation results for a use case scenario. Section 5 contains conclusions that follow from the performed research and development and finally Section 6 comprises additional discussion around the work done: practical usability of the results, their possible application areas and plans for further development.

2 Proactive Self-Maintained Resources in Semantic Web

The contribution of the ongoing SmartResource project (2004-2006) together with appropriate research effort includes prototype implementation of *distributed Semantic Web enabled maintenance management environment* with complex interactions of components, which are devices, humans (experts, operators) and remote diagnostic web-services. The environment will provide automatic discovery, integration, condition monitoring, remote diagnostics, cooperative and learning capabilities of the heterogeneous resources to deal with maintenance problems. Maintenance (software) agents will be added to industrial devices, which are assumed to be interconnected in a decentralized Peer-to-Peer network and which can integrate diagnostic services in order to increase the maintenance performance for each individual device. The maintenance case is expected to demonstrate the benefits and possibilities of new resource management framework and Semantic Web technology in general. An approach to that case harnesses the potential of emerging progressive technologies, such as Semantic Web, Agent Technology, Machine Learning, Web Services and Peer-to-Peer.

---

[1] Official website of the SmartResource project, http://www.cs.jyu.fi/ai/OntoGroup/SmartResource_details.htm
[2] Official website of the Industrial Ontologies Group, http://www.cs.jyu.fi/ai/OntoGroup/

2.1. The Background Concept: a Global Understanding Environment

Global Understanding Environment (GUN) [33] is a concept used to name a Web-based resource "welfare" environment, which provides a global system for automated "care" over (industrial) Web-resources with the help of heterogeneous, proactive, intelligent and interoperable Web-services. The main players in GUN are the following resources: service consumers (or components of service consumers), service providers (or components of service providers), decision-makers (or components of decision makers). All these resources can be artificial (tangible or intangible) or natural (human or other). It is supposed that the "service consumers" will be able: (a) to proactively monitor own state over time and changing context; (b) to discover appropriate "decision makers" and order from them remote diagnostics of the own condition, and then the "decision makers" will automatically decide, which maintenance ("treatment") services are applied to that condition; (c) to discover appropriate "service providers" and order from them the required maintenance. Main layers of the GUN architecture are shown in Figure1.

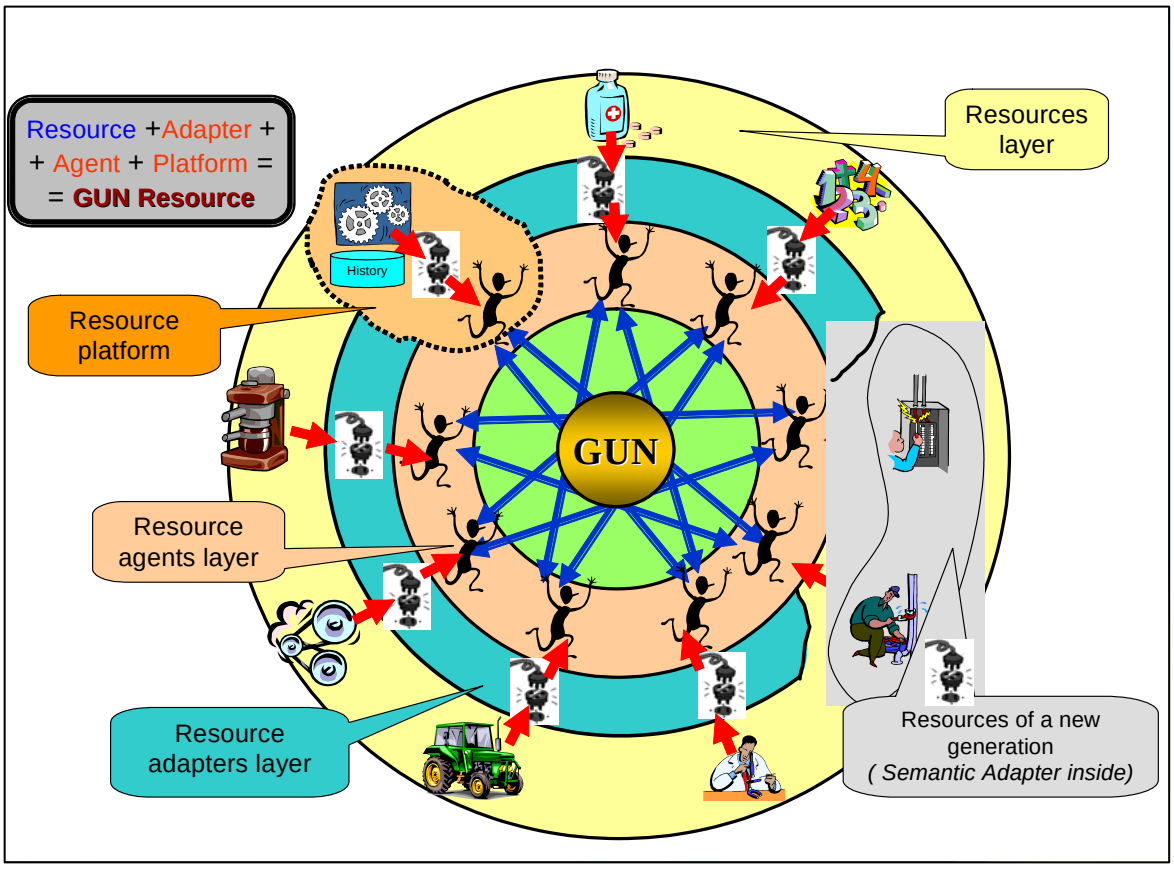

Figure 1 - Layers of the GUN architecture

Industrial resources (e.g. devices, experts, software components, etc.) can be linked to the Semantic Web-based environment via adapters (or interfaces), which include (if necessary) sensors with digital output, data structuring (e.g. XML) and semantic adapter components (XML to Semantic Web). Agents are assumed to be assigned to each resource and are able to monitor semantically reach data coming from the adapter about states of the resource, decide if more deep diagnostics of the state is needed, discover other agents in the environment, which represent "decision makers" and exchange information (agent-to-agent communication with semantically enriched content language) to get diagnoses and decide if a maintenance is needed. It is assumed

that “decision making” Web-services will be implemented based on various machine learning algorithms and will be able to learn based on samples of data taken from various “service consumers” and labeled by experts. Use of agent technologies within GUN framework allows mobility of service components between various platforms, decentralized service discovery, FIPA communication protocols utilization, and MAS-like integration/composition of services [8].

2.2. Main Stages towards Smart Resources

We have divided the implementation of the GUN concept for the maintenance domain into the following three stages: adaptation stage, proactivity stage and networking stage. Each stage assumes design of a more enhanced version of the maintenance environment.

Adaptation stage defines Semantic Web-based framework for unification of maintenance data and interoperability in maintenance system via adding explicit semantics into existing data representation formats. The semantics (metadata), which are intended to be added to the data that describe corresponding industrial resources, includes knowledge about their state, condition and diagnosis in temporal and contextual space. Further, the semantically reach resource descriptions will be used as input to decision-making components (e.g. Jess-based [10]) of software agents. The research and development tasks of this stage include development of generic semantic adapter mechanism (General Adaptation Framework) and supporting ontology (Resource State/Condition Description Framework) for different types of industrial resources: devices, software components (services) and humans (operators or experts). The key technology, which is utilized during the adaptation stage, is the Semantic Web.

Proactivity stage focuses on an architectural design of agent-based resource management framework and on enabling a meaningful resource interaction. Its research and development tasks include adding software agents (maintenance agents) to the industrial resources, enabling their proactive behavior. For this purpose, Resource Goal/Behavior Description Framework has to be designed, which will be the basis for making resource's individual behavioral model. The model is assumed to be processed and executed by the RGBDF engine used by the Maintenance Agents. Agent-based approach for management of various complex processes in the decentralized environments is being adopted and popularized currently in many industrial applications. Presentation of the resources as agents in the multi-agent system and use of technologies and standards developed by the agent research community seems as a prospective way of industrial systems development. Creation of a framework for enabling resources’ proactive behavior and such agent features as self-interestedness, goal-oriented behavior, ability to reason about itself and its environment and to communicate with other agents, can bring a value to the next-generation industrial systems.

The objective of the *Networking Stage* comprises complex behavior/interaction scenarios of Smart Resources (agent-augmented Device, Expert and Service) in a global decentralized networked environment. The scenarios assume agent-based interoperation of multiple devices, multiple services and multiple experts, which allows discovery of necessary experts in Peer-to-Peer network, using their experiences to learn remote diagnostics Web-services, making online diagnostics of devices by integrating diagnoses from several services, learning models for a device diagnostics based on online data from several distributed samples of similar device, etc. Emerging Peer-to-Peer technology and similar network architectures suite well the increasingly decentralized nature of modern companies and their industrial and business processes, whether it is a single enterprise or a group of companies [37]. The set of attractive features of the Peer-to-Peer model includes decentralization, scalability and fault-tolerance along with low

administration expenses. Client/server architectures with centralized management policy increasingly fail with big amounts of nodes, because of their complexity and extremely high demands on computing resources. Distributed content management systems address the need to access content wherever it resides, produce content while maintaining control over it, and collaborate efficiently by sharing real-time data within a distributed network of stakeholders.

This paper describes in details the results of the first mentioned stage and represents the second stage as plans for future in section 6.3. The third stage of detailed research and implementation of the GUN vision (Networking) remains as a planned perspective.

3 Essentials of a Resource Adaptation

3.1 Semantic Data Model

We cannot say yet that Semantic Web technology as such is mature enough to be accepted by industry in a large scale. The reasons for that we have analyzed in [11, 16, 34, 36, and 37] and recent prognoses state the same [5]. Some of standards still need modifications as well as appropriate tool support. For example, Semantic Web technology offers a Resource Description Framework (RDF) as a standard for semantic annotation of Web resources. It is expected that Web content with RDF-based metadata layer and ontological basis for it will be enough to enable interoperable and automated processing of Web data by various applications. However emerging industrial applications consider e.g. machines, processes, personnel, services for condition monitoring, remote diagnostics and maintenance, etc. to be specific classes of Web resources and thus a subject for semantic annotation. Such resources are naturally dynamic, not only from the point of view of changing values for some attributes (state of resource) but also from the point of view of changing "status-labels" (condition of the resource). Current RDF still needs temporal and contextual extensions [25].

This motivates one of the objectives of SmartResource activities during the Adaptation Stage, which is Resource State/Condition Description Framework (RSCDF), as an extension to RDF, which introduces upper-ontology (semantic standardized data model) for describing such characteristics of resources as states and corresponding conditions, dynamics of state changes, target conditions and historical data about previous states. These descriptions are supposed to be used by external Web-services (e.g. condition monitoring, remote diagnostics and predictive maintenance of the resources). Pilot version of RSCDF and appropriate schema developed using the freeware open source Protégé[3] tool, are presented in [13]. Querying specifics of RSCDF was analyzed in [25]. Figure 2 depicts the conceptual meaning of RSCDF: it is an RDF-compliant semantic representation format for resource's historical (life-cycle) data.

RSCDF inherits from RDF an approach of modeling a problem domain utilizing inter-related hierarchies of classes and properties. Special emphasis in RSCDF is made on context-sensitive semantic description of Web resources. This approach endows the resulting data models with high extensibility and originally aims at providing a semantically rich descriptive data (metadata) about a corresponding resource to a new-generation (intelligent) software processing tools.

[3] http://protege.stanford.edu/, official website of the Protégé tool

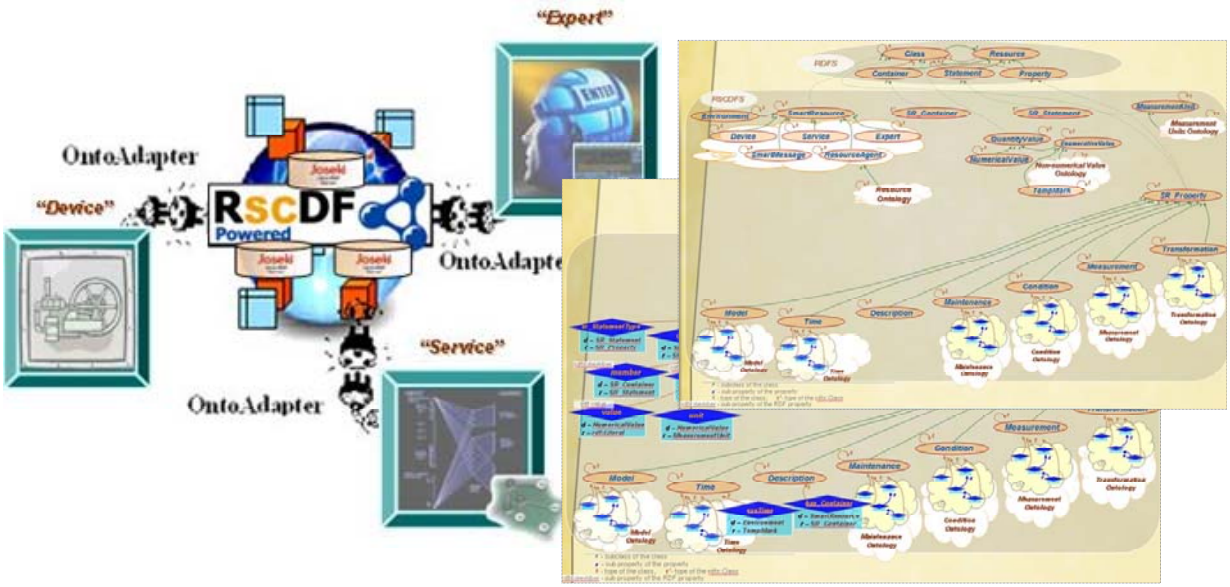

Figure 2 - Conceptual meaning of the Resource State/Condition Description Framework

To utilize the RSCDF advantages in other than industrial maintenance domains, a simplified version of RCSDF called Context Description Framework (CDF) [15] has been designed, which includes only following basic new components comparably to RDF: (1) contextual representation of RDF triplet statement (subject-predicate-object-context), where the context is represented with a container of RDF statements; (b) the definition of a property in RSCDF Schema in addition to definition of a domain and a range will also include the definition of a context of the property as the set of possible properties from the context container for this property (see Figure 3).

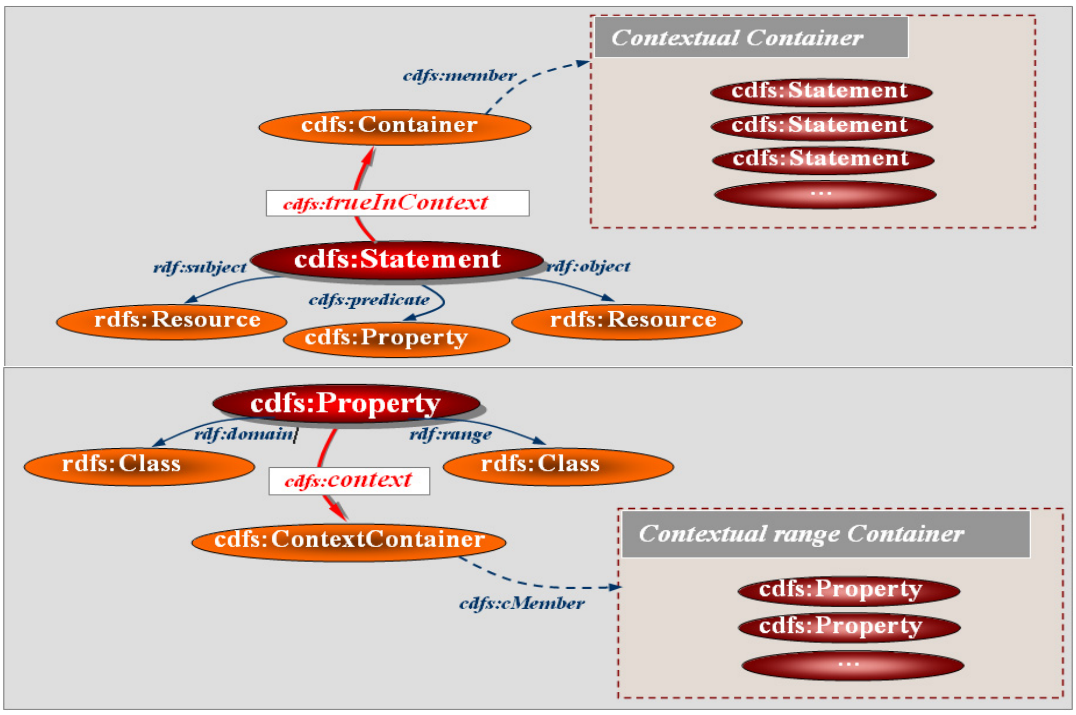

Figure 3 - Basics of the RSCDF extension in comparison to RDF

3.2 General Adaptation Framework

Another obstacle for the Semantic Web standardization effort relates to the fact that despite many industrial companies and consortiums have realized that explicit description of semantics of data and domain modeling is necessary for application integration, they have still used for that purpose their company/consortia specific standards or XML language that is inappropriate for global integration. Even realizing that Semantic Web is providing really global standards, it is already too late, labor and resource consuming to transform manually huge amount of already modeled metadata from a local to the global standard. One possible solution could be designing semantic adapters, which enable semiautomatic transformation from company specific standards to Semantic Web standards. This motivates the second objective of the SmartResource Adaptation Stage, which is a design of General Adaptation Framework aimed to provide a methodology for designing adapters from various data formats to RSCDF and back. The pilot version of the task and its solution are presented as deliverables of the SmartResource project[4] along with concrete test implementations of the approach of General Adaptation - adapters for three different samples of heterogeneous resources (device data, expert interface, Web-Service). The conceptual picture of General Adaptation Framework is shown in Figure 4.

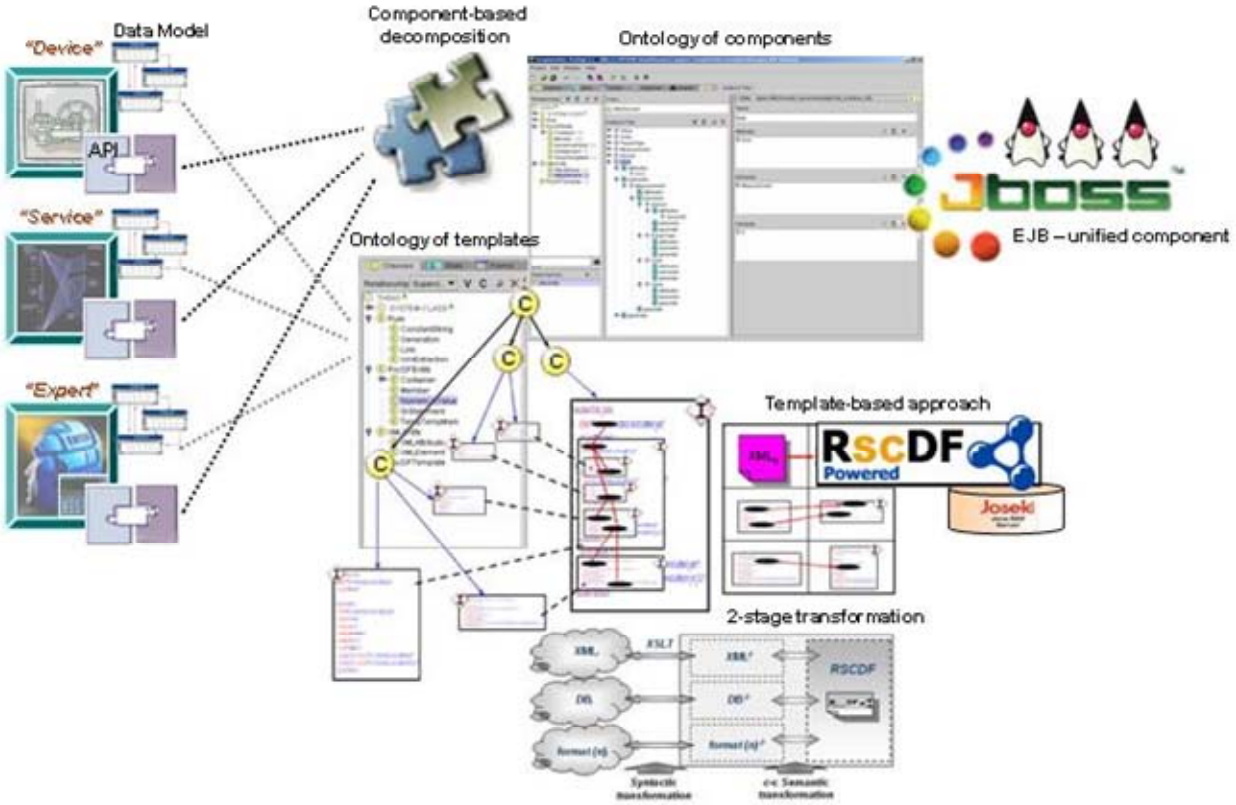

Figure 4 - General Adaptation Framework illustrated

[4] Web pages of Deliverables 2 and 3 of the SmartResource I project, http://www.cs.jyu.fi/ai/OntoGroup/Deliverable2.htm, http://www.cs.jyu.fi/ai/OntoGroup/Deliverable3.htm.

In the approach of General Adaptation we distinguish two aspects of adaptation: data model transformation and Application Programming Interface (API) adaptation as it was mentioned in Introduction. The first aspect focuses on a transformation of resource data stored in a specific data model (relational database, family of XML-based standards, UML, etc.) to a unified semantically-rich format, in our case to RSCDF, and vice versa. For this purpose, we utilize a method of two-stage transformation, which assumes mapping of a specific data model to a corresponding canonical form from the same family of data representation standards. If, for instance, we need to transform an XML schema to RSCDF, first of all we have to define the XML canonical schema and make a mapping to it.

The strength of the two-stage transformation is in reuse of a variety of existing powerful tools for data model mapping[5] and also in simplification of the data model mapping process for potential customers – owners of resources that are intended to be integrated into the target maintenance environment. The owners do not have to think about complicated ways of transformation of their data models to RDF-based standards – they just have to map their data model to the canonical one within the same standard (e.g., XML). After native-to-canonical data model mapping, the template-based approach of semantic transformation from a canonical form to RSCDF is applied according to GAF. This approach is based on automated generation of XML serialized RSCDF instances, which are determined from the ontology of templates. The ontology stores classified pairs of correspondence between canonical and RSCDF patterns – chunks of terminal strings of text. Thus, in fact, thanks to GAF, the process of data model transformation requires only two relatively simple manual efforts: mapping between the initial and canonical data schemata, and engineering of the ontology of templates. Having these two activities done, the data transformation between a native and RSCDF formats is carried out automatically.

The second aspect of adaptation (API adaptation) relates to a possibility of automated access to data entities in native storages through native application interfaces. For instance, a database entity can be accesses via ODBC (Open Database Connectivity) connectors using functional calls in different programming languages. To access a certain database records for further data transformation an appropriate programming component must exist. The component can either execute native functional calls or perform a direct access to the native data storage. Hence, to automate the retrieval of native data entities the existing types of API's must be decomposed using component-based analysis [23, 24 and 38], classified and arranged into a centralized/decentralized library [18]. Such components, in a vision of GAF, are building blocks for automated assembly of concrete adapter "on the fly". The automated component integration is performed using ontology of components, and the resulting adapter is run as an EJB[6] (Enterprise Java Bean) component on a JBoss Application server[7] in our implementation.

To a have a comprehensive framework for adaptation of resources, ontology of templates and ontology of components must be closely interrelated due to high dependency between data models and methods of accessing the corresponding data.

3.3 SmartResource Prototype Environment

For a practical testing of the developed General Adaptation approach, the first version of the target prototype environment has been implemented. The environment can be launched on one or

[5] MapForce homepage. http://www.altova.com/products_mapforce.html
[6] http://java.sun.com/products/ejb/, Enterprise JavaBeans Technology
[7] http://www.jboss.org/products/jbossas, description of the JBoss Application Server

several workstations, which meets the specified installation requirements. Figure 5 illustrates architecture of the implemented prototype environment.

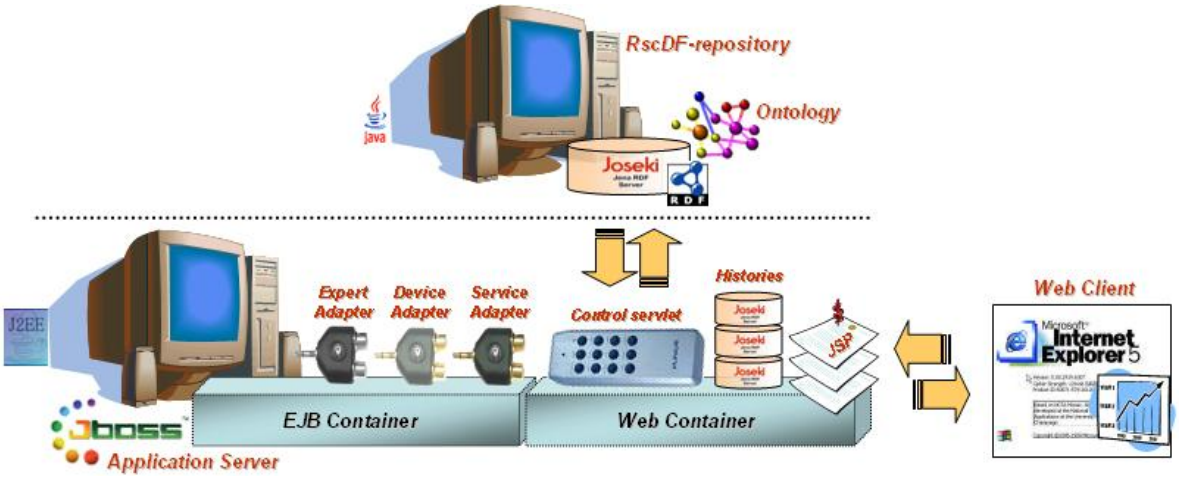

Figure 5 - Architecture of the SmartResource prototype environment, v. 1.0

For the process of software engineering the latest and the most powerful freeware and open source tools and technologies have been used. The whole environment is based on Java 2 Platform, Enterprise Edition[8] (J2EE) and was developed using Eclipse[9] Integrated Development Environment together with the Poseidon[10] UML-based modeling tool. Versioning control was carried out with help of the CVS[11] tool. As it was mentioned, for testing the approach of General Adaptation Framework and the RSCDF format, three sample adapters were implemented (for a device, an expert and a Web-service). Their logic was encapsulated in three Enterprise Java Beans (EJB) and executed on the JBoss application server. Specification of the KF-330 Blow Molding Machine was used for simulation of the device data (7 device parameters). Device states were generated in a form of XML entities according to the corresponding XML schemata (three different schema variations, plus a canonical one). State and Condition resource data have been encoded in RSCDF after the transformation and stored in a remote Joseki[12] RDF server. For creation of a local history cache, Jena[13] classes were used. Code that coordinated coherent work of the adapters and provided a control/monitoring over them, was executed in the control Java Servlet[14]. Visualization of the internal processes of the prototype environment was organized using a set of Java Server Pages[15] (JSP). Demonstrations were carried out using Internet Explorer web browser.

Web Service adapter incorporated a simple sample of learning algorithm (KNN-method) wrapped by a web service container using Axis[16] and Lomboz[17] (see Figure 6). The adapter using

[8] http://java.sun.com/j2ee/, Java 2 Platform Enterprise Edition
[9] http://www.eclipse.org/, Eclipse Integrated Development Environment
[10] http://www.gentleware.com/, Poseidon UML modeling tool
[11] https://www.cvshome.org/, CVS – Concurrent Versions System
[12] http://www.joseki.org/, Joseki RDF server
[13] http://jena.sourceforge.net/, Jena – A Semantic Web Framework for Java
[14] http://java.sun.com/products/servlet/, Java Servlet Technology
[15] http://java.sun.com/products/jsp/, JavaServer Pages Technology
[16] http://ws.apache.org/axis/, official webpage of Axis Apache
[17] http://www.objectlearn.com/index.jsp, Lomboz ObjectLearn Eclipse plugin

generated SOAP-client simulated software agent's requests for learning and diagnostics. For RSCDF-XML transformations, the adapter uses approach of 2-stage transformation with RDQL-templates.

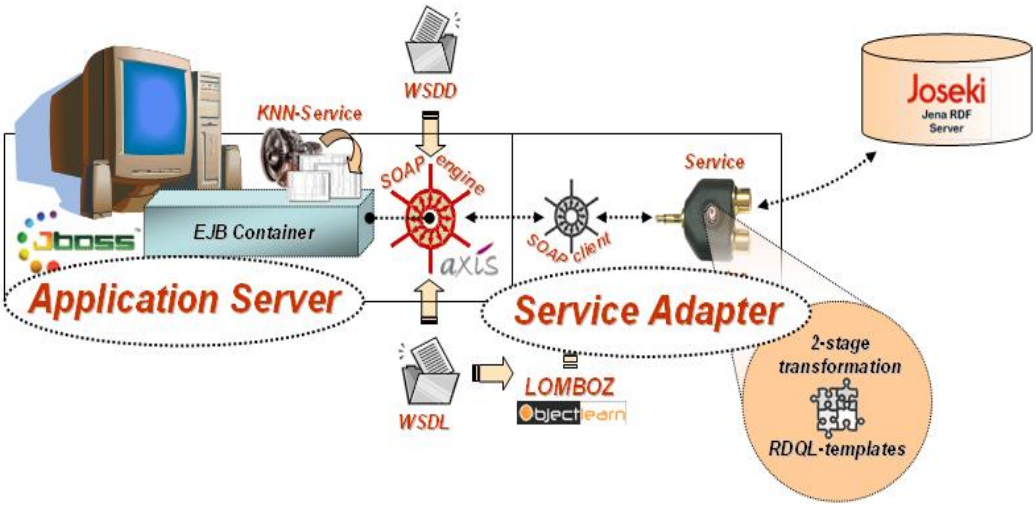

Figure 6 - Implementation architecture of the Web Service adapter

In the implementation of the human expert adapter, 2-stage transformation and User Interface Templates were used for flexible building of a specific human interface (Figure 7). Involvement of the JFreeChart[18] open Java library allowed generating images for representation of the device states. Human Expert is requested for a diagnostics via e-mail.

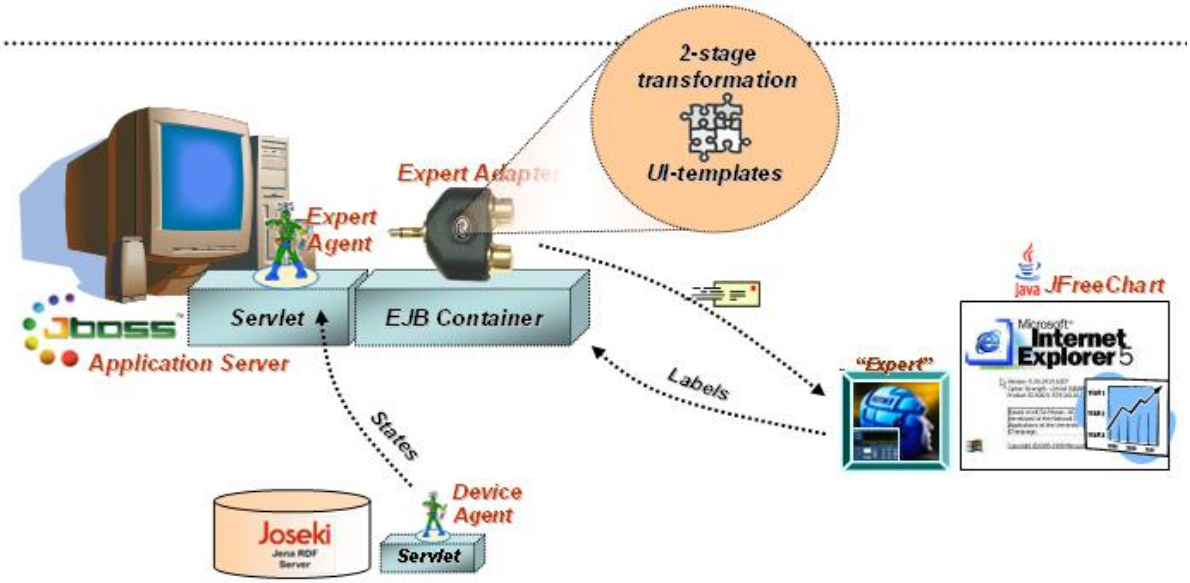

Figure 7 - Implementation architecture of the human expert adapter

[18] http://www.jfree.org/jfreechart/, JFreeChart – free Java class library for generating charts

4 Implementing General Adaptation Framework

4.1 Challenges

There is a variety of resources intended for integration into a common SmartResource environment. For more efficient analysis, all resources were divided into three basic classes: devices, services and humans. These resources represent real world objects, which should interact in certain way according to appropriate business models. The adaptation of such heterogeneous resources in common sense lies in providing an environment, which would allow them to communicate in a unified way via standard protocol.

The primary intention behind the General Adaptation Framework (GAF) is a design of common framework for adaptation of heterogeneous resources. The design of the framework will be divided into two layers:

1. Structured software design for modules, classes, behavior and protocols;
2. Semantic adaptation of different formalizations of the problem domain edges.

A *semantic transformation* is one of the key problems in development of the General Adaptation Framework. We assume that semantic annotation of data, which are used in communication between heterogeneous software components based on common ontology [9], will enable interoperability [19].

At the moment, arbitrary number of standards exists, which define each other on different levels of abstraction and thus form a hierarchy. There are quite many data models and one of them, which have recently gained wide adoption, is XML (Extensible Markup Language). The older and more tested data representation standard is Relational Model. The novel data representation standards, which focus primarily on semantics, are RDF and OWL. All these mentioned data representation standards have to be analyzed foremost, to understand the essence of semantic transformation. As we can see, the standards provide specifications as guidelines to formalization of various problem domains. For a concrete problem domain the necessary schemata are constructed as a formalized domain models based on the corresponding specifications. Content (documents, database records, any structured data) that include set of facts within the chosen problem domain, are structured according to the developed schemata and specifications. More abstract models define the more specific ones. In different cases arbitrary number of models can be found in chains and layers. In this perspective, the semantic transformation results to extraction of data semantics independently from particular data representation standard. This approach must be used to allow encoding of these data to another representation standard without loosing of the meaning.

During the semantic transformation process, transforming object/module involves format's metadata (schemas) and transformation rules. Schemas, rules and underlying ontologies constitute a framework for semantic transformation. Semantic transformation defines a functionality to work with semantics of:

- Adapter functionality (services provided by an adapter);
- Data representation standards and models of adapter systems;
- Software interface standards of adapted systems;
- Configuration properties of an adapter runtime environment.

Given that unambiguous semantic description of resources supposed to be machine-processable, an automated adapter composition will be possible. However, unambiguous

semantic description requires a human to map the meaning of concepts and relations unless this mapping already exists in some ontology. The tools will be needed to simplify the process of mapping for human [12]. Tools will use faceted classification, adapted for each particular domain in order to make the most relevant concepts easily accessible.

The following cases are essential in the context of automated semantic adaptation:

1. Explicit mapping (human assisted).
2. Shared ontology (both resources are mapped to the same ontology).
3. Shared ontology lookup & composition (may be wrapped as a service or implemented as an embedded functionality).

4.2 Pilot Implementation of Semantic Adapters

4.2.1. Two-stage transformation

The SmartResource domain needs RDF and its RSCDF extension as a basis for the formalization. Also necessary concepts definitions are included into RSCDF-schema (Resource State/Condition Description Framework, see detailed description in [13]). To meet the challenge of semantic adaptation, an ontology based approach is used to define the semantics. This involves associating a commonly used meaning to the definition of adapter properties, functionality, configuration, and corresponding meta-data standards.

The SmartResource addresses the adaptation challenge using the *two-stage transformation*:

- Syntactical transformation;
- Canonical-to-canonical semantic transformation.

Such technique seems reasonable because the division into two independent phases facilitates the whole transformation process. This is possible in cases, when tools for (semi)automatic syntactical transformation exist. A specific canonical form for a given domain description should be available for every data representation standard so that a transformation is performed between different schemas of the same data model.

XSLT-based transformation is a good example of syntactical transformation of XML files. In our case, XSLT is used for syntactical transformation between different XML-schemas (XPATH expressions are also a possible solution). Each document of a certain standard (for XML it is XML_i) is transformed into a corresponding canonical form (for XML, it is XML^0) during the syntactical transformation stage, as it is shown in Figure 8. During the second stage (canonical-to-canonical semantic transformation), the canonical form (e.g. XML^0) is transformed into the unified semantic canonical form, which is RSCDF in our case.

The two-stage transformation assumes functioning in both directions. That is, the RSCDF-XML^0-XML_i path of the transformation is equally in the scope of the analysis. There are few projects, which have elaborated pilot methods of transformation RDF to XML [22, 42]. Since RSCDF is an enhanced subset of RDF, it is possible to adopt these methods also.

Once the mechanism of transformation from RSCDF to XML and XML to RSCDF has been designed, it is possible to use standard approaches for future transformations (to other existing standards). We assume such approach to decrease the complexity of whole transformation task because existing tools and standards of syntactical transformation can be reused and utilized. Canonical form limits variety of syntactical representation of the same domain to a strict syntactical form and allows template based approach for semantic transformation.

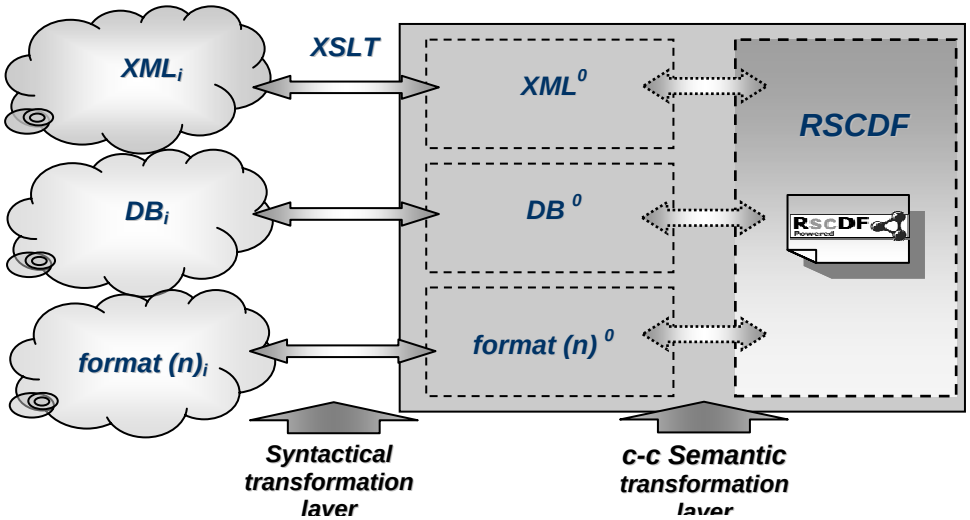

Figure 8 - Two-stage transformation

From the existing commercial tools that provide transformation of XML to other formats, Altova MapForce can be mentioned[19]. This commercial tool allows XML to XML transformation based upon two XML schemas (Figure 9, left picture). It also might be necessary to perform some processing functions to pipe data from source to target.

MapForce allows mapping between XML and Relational database, too (Figure 9, right picture). The process of mapping starts from the loading of database schema and XML schema. Then engineer manually fulfills matching between XML elements and database entities. While mapping, it might be necessary to use processing functions.

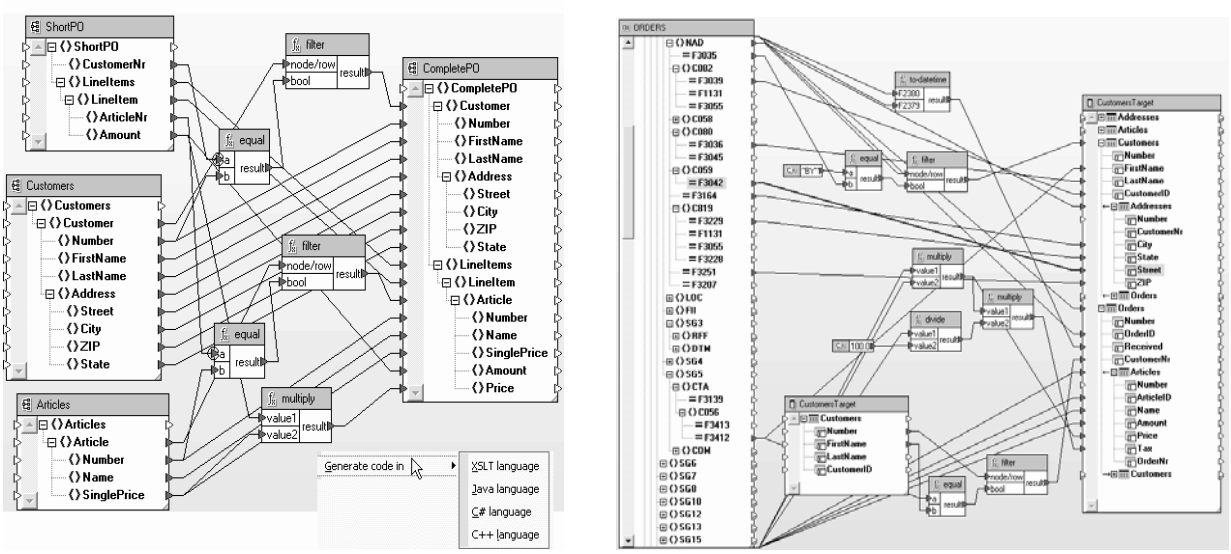

Figure 9 - XML transformation in MapForce, adopted

[19] MapForce's homepage. http://www.altova.com/products_mapforce.html.

4.2.2. Canonical forms

The development of the canonical forms for particular problem domain involves domain experts and takes into account existing formalizations of the same problem domain. As an example of the latter, Paper IXI[20] – a consortia wide XML-based standard for Paper Mill model can be mentioned.

The domain, which is in focus of the SmartResource activities, is related to the paper industry, paper machines and a process of paper manufacturing. The first stage of the development of canonical form for this domain will be elaboration of a conceptual model for it. Firstly, the domain description in a natural language must exist. It can be made either separately, or existing specifications can be used. The main point is that this description must contain all important aspects of the problem domain. For our domain, the description can include such phrases as "a paper machine produces paper, uses cellulose", etc.

The domain decomposition follows the domain description and is based on it. On that stage, entities, classes, properties, relations, behaviors of the problem domain are distinguished. After the necessary decompositions the domain formalization is performed using any appropriate data models. It can be ER-diagrams (Entity Relationship), UML, Ontology, etc.

Then analysis of data representation format, which will be used for the canonical form, should be performed. It includes analysis of the data format type (XML, text file, Excel table, Oracle database, etc.), types of APIs that can be used in the domain (SQL-queries, Java DOM API, XQuery, etc.), access methods to data (JDBC, OLE, etc.), sorts of standards that are used to represent a format (ASCII, W3C-family standards).

The first stage of the canonical-to-canonical semantic transformation is a metadata analysis. This stage includes analysis of data schema used in the canonical form (elements, relationships, types, etc.), possible variations (XML tags or values, etc.), hierarchy of elements and restrictions (nesting of classes, range, etc.).

Further stage of the canonical-to-canonical semantic transformation is analysis of standard that has been chosen for the canonical representation form. This stage includes analysis of standard specification (syntax, vendors, schema, etc.), analysis of existing formal theory (relational algebra, frame model, etc.), analysis of existing methods of transformation (XSLT, production rules, etc.), analysis of capabilities and restrictions (possibilities of formalization, querying, etc.).

The final step of the canonical-to-canonical semantic transformation is concerned with data mapping rules. This paper considers a use case of XML-RDF transformation. This stage requires efforts for determining a protocol of transformation (elements and types matching), representation format for the rules (Ontology, XSLT, etc.), percentage of manual, semiautomatic and automatic matching actions.

In the SmartResource pilot implementation, according to the approach of two-stage transformation, canonical XML schema was designed and other three different XML schemata were used for testing the phase of syntactical transformation. Those three schemata were dedicated to describe the same semantics (physical measurements), but using different structural organization and different syntactic elements (XML attributes and tags). The canonical schema is designed as incorporating a unification of all semantically significant XML tags and representing a single syntactical option for them. To perform syntactical transformation to the common XML canonical form, for each of the three XML schemata corresponding XSLT files were generated

[20] Official website of the PaperIXI project, http://pim.vtt.fi/paperixi/.

using MapForce trial version. Figure 10 contains fragments of one XML_i file, corresponding $XSLT_i$ used for transforming and the fragment of resulting XML file in the canonical form.

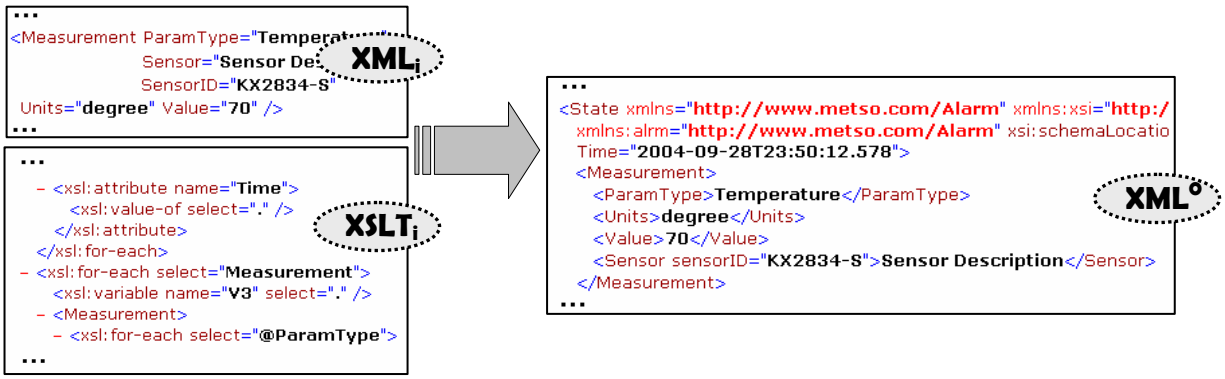

Figure 10 - Example of syntactical transformation

The mechanism of transformation requires the following analyses to be done: analysis of possible approaches (tools, APIs, Services, etc.), estimation of cost for particular approach (time for development, price of the product, etc.), study of interoperability and extensibility of the chosen approach (supported platforms, extensible API, etc.). For transformation, existing tools can be used or if reasonable these tools can be developed from scratch. The most popular APIs used in transformation of XML are XSLT, SAX and DOM. In case of RSCDF the functionality for implementation must be defined: either it will be XML-to-RDF transformation, or more.

4.2.3. Use Case Scenario

Since many details about the SmartResource pilot implementation have been covered in Section 3, here we give just an example of the whole cycle of adaptation that takes place in the pilot environment. Some implementation details that have not been mentioned above are given also.

The use case scenario that is used for testing the pilot implementation is based on the interaction procedures between heterogeneous Device, Service and Expert (see Figure 11).

The scenario includes device diagnostics by a human expert, which watches the device history through the expert adapter and puts diagnostic labels on the device states after analysis. The labeled data in the RSCDF format is stored in the history of the device and further is used for learning procedure with the service. Service, to be able to read meaningfully the device history, utilizes the corresponding adapter.

Thus, the tasks of the three adapters generated using General Adaptation Framework, are:

1. Transform XML-based descriptions of the device history into the appropriate RSCDF form (instances).
2. After that on the request of the expert (diagnostics) or service (learning/diagnostics), adapter has to transform device data from the RSCDF form to the representation more convenient for the expert/service. Feedback of the expert or of the service has to be converted again into RSCDF for further reading by the device logics.

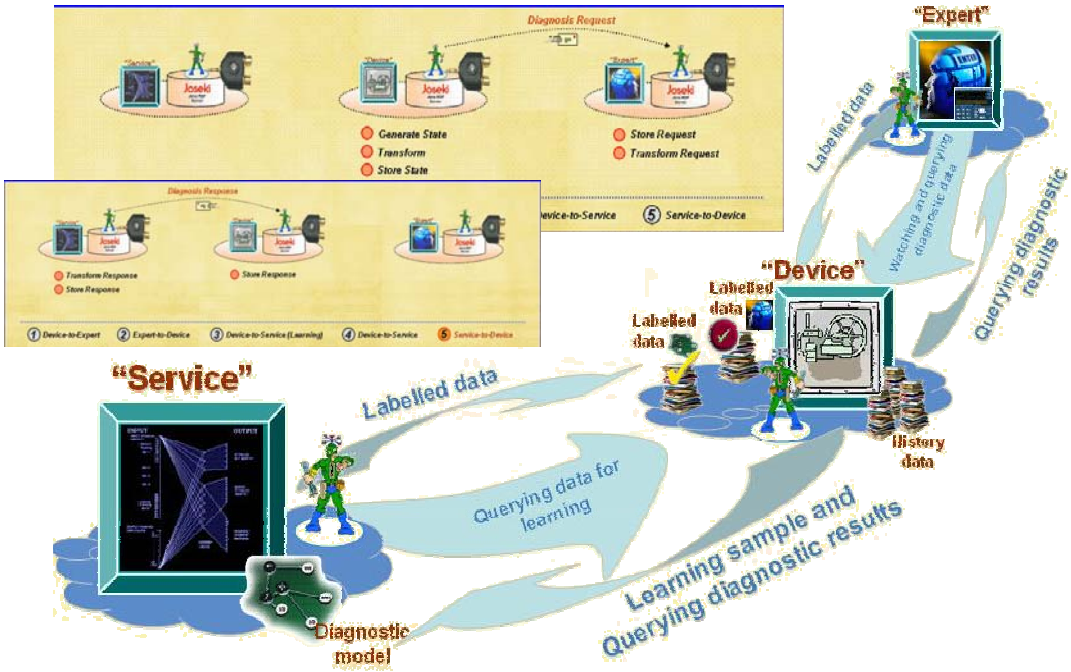

Figure 11 – Use case scenario of the SmartResource I

Let us describe the sequence of operational steps that occur according to the use case scenario. The corresponding sequence diagram created with the Poseidon UML modeling tool is presented in Figure 12. In the figure, four acting objects of the scenario are shown. They are: Service Prototype, Device Prototype, Expert Prototype and additionally a Human User actor. The prototypes comprise corresponding adapters and a simple logic of interaction between each other is used for testing the adapters. The Human User actor represents web-browser based user interface designed for monitoring and controlling the functionality of the adapters. The user interface is implemented as HTML image maps (couple of them can be seen in Figure 11).

The use case scenario comprises five interaction phases: (1) Device-to-Expert, (2) Expert-to-Device, (3) Device-to-Service (Learning), (4) Device-to-Service (Diagnostics), (5) Service-to-Device. Each of them tests a concrete functionality of the adapters and interoperability between the underlying heterogeneous components. In the sequence diagram, just 1-3 interaction phases are included, because 5 and 6 duplicate the dialog of the device with expert (1 and 2). Each interaction phase is divided into a sequence of stimuli that denote atomic interactions between the actors of the diagram. Each stimulus has a name pattern *stim_X.Y:* followed by the name of atomic interaction, where X denotes a number of the interaction phase and Y – number of the atomic interaction. For example, *stim_1.2:Transform* denotes a stimulating request sent by a user for invocation of the logic of the Device Adapter. In this paper we show stimuli from the sequence diagram that relate to the Device Adapter. The first one is *stim_1.1:ReturnState*, which denotes a process, when Device Prototype returns a generated device state to the user in a form of chart that has underlying XML (canonical form mentioned Section 4.2.2) representation (see Figure 13).

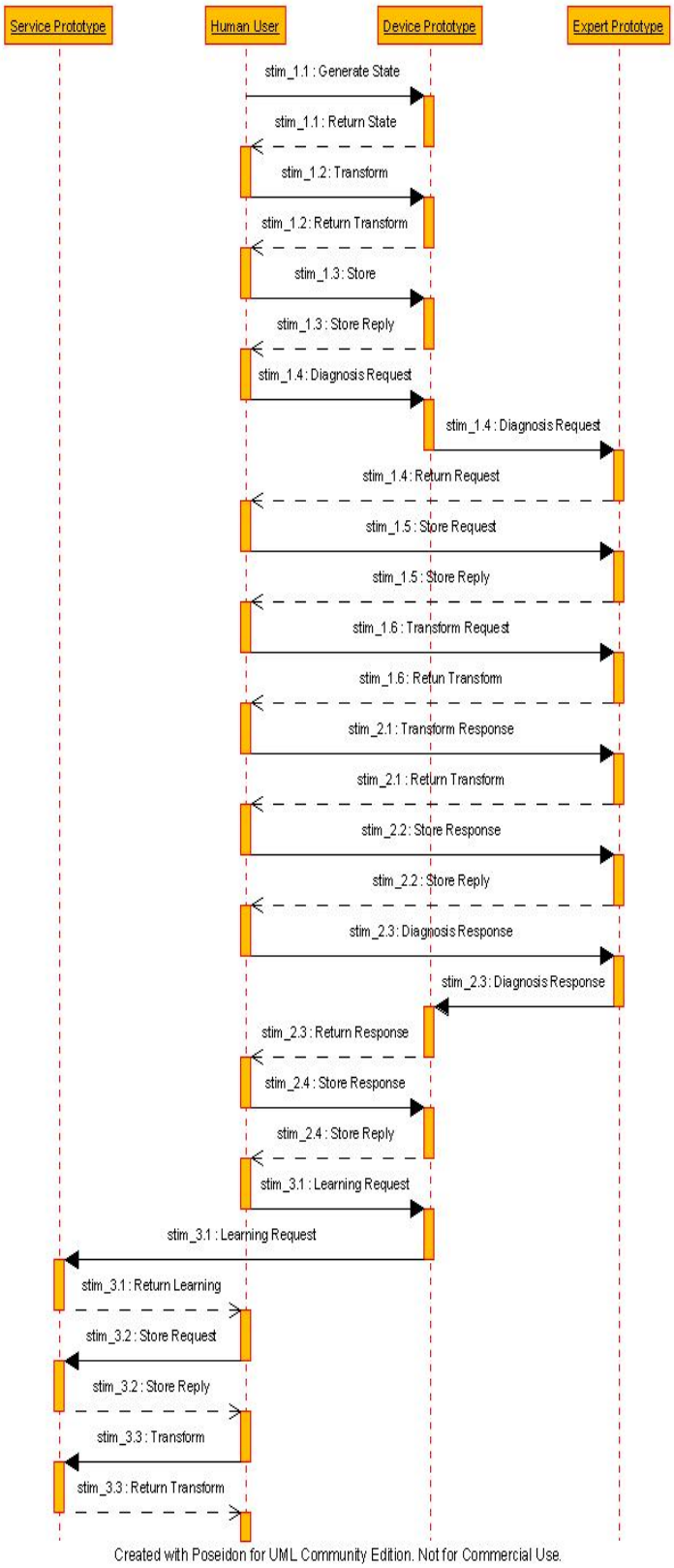

Figure 12 - Sequence diagram of the *SmartResource I* use case scenario

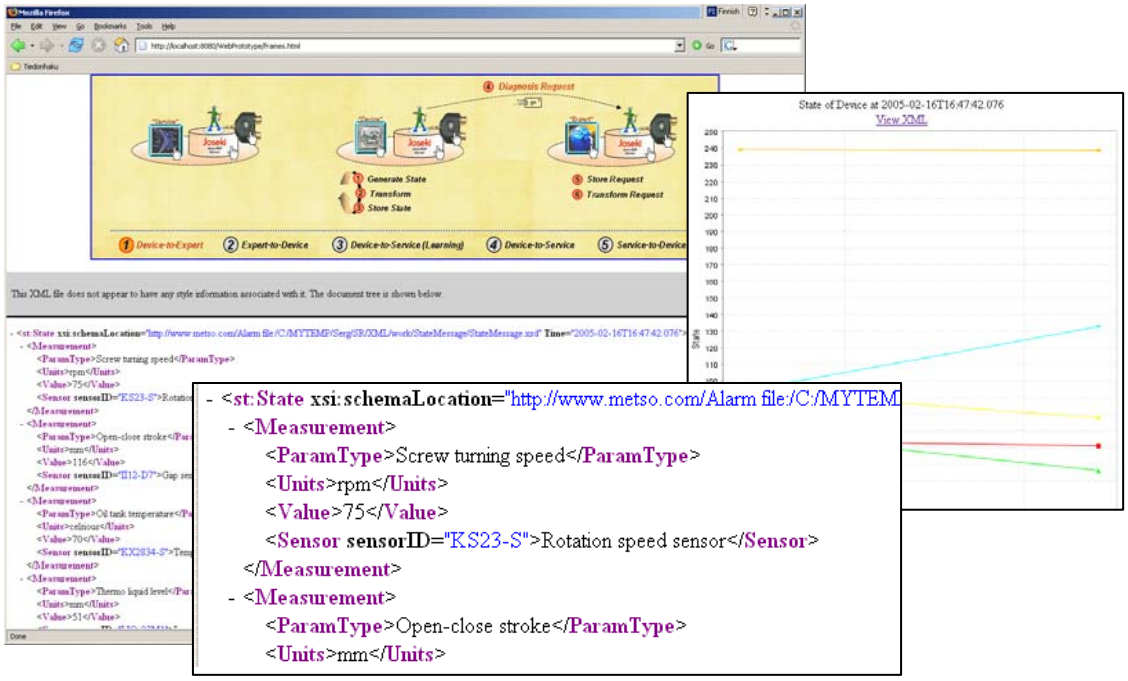

Figure 13 - Device state represented in XML and in a chart

Stimulus *stim_1.2:ReturnTransform* is also worth mentioning, because it reflects sending results of the transformation process performed by the Device Adapter to the user in a form of RSCDF instances. All classes, which constitute the device adapter, are packaged into one template package (see Figure 14). Logically the classes could be divided into four parts. The first part of classes corresponds to the logic, which reflects the structure of the RSCDF document, the second one reflects the structure of the original XML document and encapsulates the logic of processing this structure, the third one represents the engine which plays the role of RSCDF document builder, and the fourth one is the set of reusable utilities for DOM processing.

For implementation of the second phase (XML^0-RSCDF canonical-to-canonical semantic transformation), the method based on templates was applied. During the analysis of RSCDF document, reusable templates can be extracted. For instance, from a RSCDF document two types of templates were distinguished: structural templates (or patterns) and tag templates. Structural templates reflect the structure of the RSCDF graph according to its schema. Depending on a canonical XML document, some branches of the RSCDF graph have the same structure and could be cloned while processing.

On the other hand, the tag templates correspond to the RSCDF classes. In fact, tag template represents some classes from RSCDF schema. There are for example, *SR_Statement*, *Context_SR_Container*, *SR_Container*, *NumericalValue*, *TempTempMark*. Tag templates are “bricks”, which are used by adapter to produce the RSCDF document. Tag template has a body and a changing part, which can have different types:

- link to other tag template;
- link to XML data;
- link to ontology data;
- generated value.

Figure 14 – UML diagram of classes for a semantic adapter of a device

Figure 15 represents an example of the tag template. The variable X_n is obtained during a run-time either from ontology or from the XML file, or generated by the generator. The variable Y_n is obtained from the identifier of from some other template. This means that if an RSCDF tag depends on some other RSCDF tag, then it will be generated after the latter one. In this way the adapter recursively calls methods of template creation until it will reach the leaf nodes.

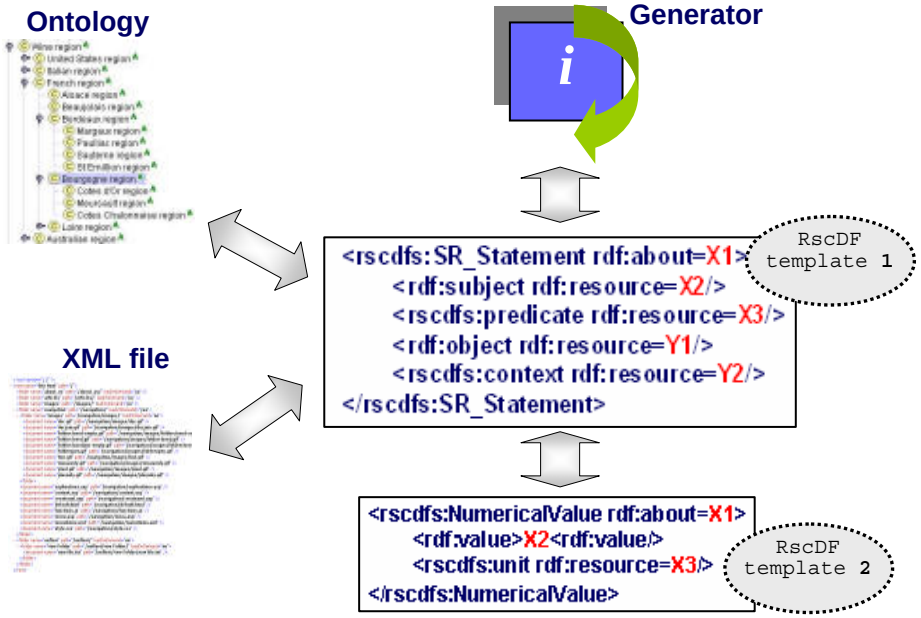

Figure 15 - Template-based transformation

Using tag and structural templates, the device adapter performs semantic transformation. This approach provides possible way to implement the logic of semantic transformation from canonical XML to RSCDF format.

For modularity of the approach and for a possibility of easy modeling over templates all necessary lexical concepts, which further will be used in manual or semi-automated (in perspective) manipulation are defined in ontology (see Figure 16).

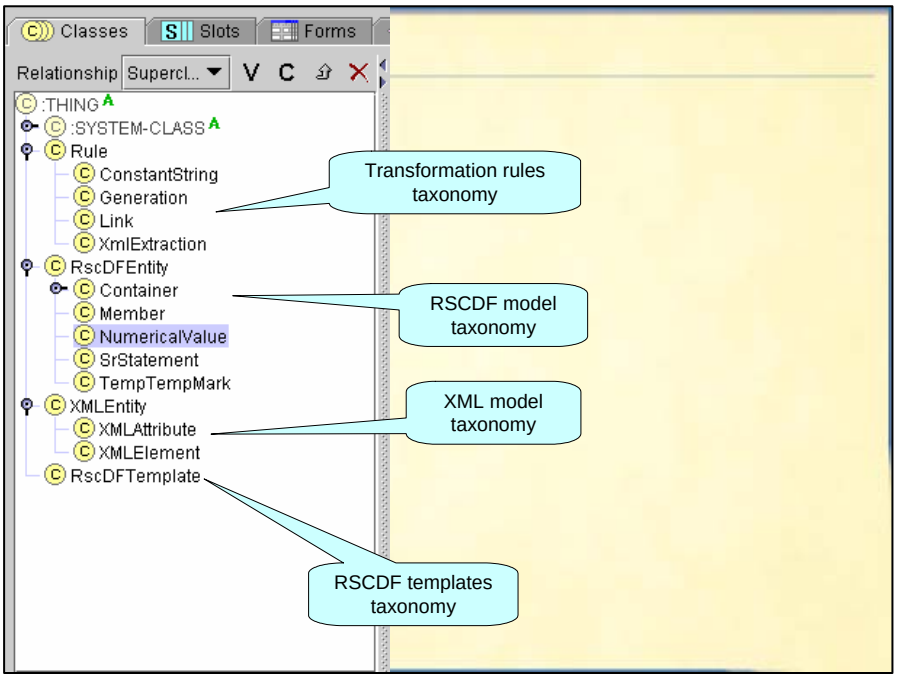

Figure 16 – Sample ontology of templates designed in Protégé 2000

Figure 17 presents an example of a template for NumericalValue concept from the ontology with defined XML serialization form, which can be reused in semantic (canonical-to-canonical) XML-RSCDF transformation.

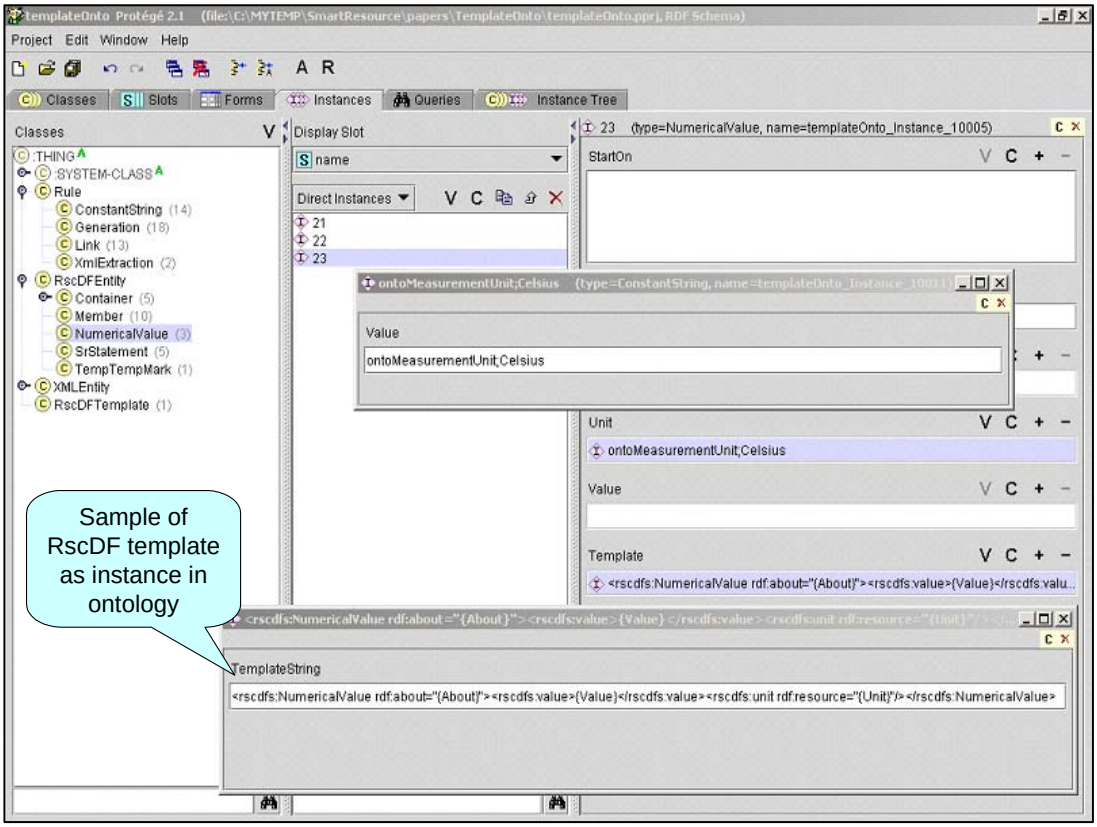

Figure 17 – Example of a template definition in Protégé 2000

5 Conclusion

The research efforts have been made to explore the potential of the emerging and promising Semantic Web technology in a challenge of adaptation of existing widely adopted models of data representation to emerging RDF-based ones. In contrast to ICT, the semantic technologies represent meanings separately from data, content, or program code, using the open standards for the Semantic Web. They are language neutral, machine interpretable, sharable, and adaptive, allow ontology based integration of heterogeneous resources.

During the Adaptation Stage that has been described in details in this paper, we succeeded in implementation and prototype testing of the semantic modeling approach for the challenge of semantic adaptation of conventional data representation models (taking XML as an example) to the upcoming RDF-based models. The ultimate goal is to develop efficient semantic modeling methodology in order to simplify the relevant problem of integration of legacy systems that are currently used to manage digital aspects of enterprise resources, to automated, agent-based environment. Currently, this problem has been attacked in its two major points: data model transformation and integration of Application Programming Interfaces. The latter remains

unexplored, but application of semantic modeling with design of necessary ontology of components, the main granules for the composition processes, has become an indubitable direction for the efficient solution. The novel combination of the semantic modeling (ontology of components) with the component-based decomposition (the latter has developed for over ten years) is assumed to increase significantly the efficiency of the Component-Oriented Analysis towards automation of the Enterprise Application Integration.

The introduction of the semantic modeling element (ontology of data model templates – structural patterns and interrelations between them) into the process of data model transformation is anticipated to be a basis for tools that will allow an automation of the transformation. Also, the competent decomposition of the data model transformation process into two stages (mapping of a native format to a canonical one and mapping of the canonical model to a semantic one) will relief resource providers of the complexities with semantics (facilitating the adoption of the Semantic Web technology) and will harness existing model mapping tools.

As for the semantic format based on the developed Resource State/Condition Description Framework, its presentation is intended to initiate a useful enhancement of the RDF standard in regard to its applicability to highly dynamic resource maintenance environments. Current version of the RSCDF schema, which contains contextual and temporal extensions, and also adapters from and to RSCDF format are assumed to be a good case to facilitate the Semantic Web technology industrial adoption.

6 Discussion and Future Work

One of the goals of the SmartResource activity is a demonstration of the benefits and possibilities that the Semantic Web technology can potentially bring to the industry. Trend within worldwide activities related to Semantic Web definitely shows that the technology has emerging growth of interest in both academia and business during quite small time interval. The stage of the technology (according to highly qualified expert evaluations [7]) is called now “From skepticism and curiosity to enthusiasm: People are now asking “How” questions as opposed to “Why” and “What””. Moreover prognoses [7] show that “semantic solutions, services and software markets will grow rapidly topping $60B by 2010”. Semantic technologies are building blocks of the next mega-wave of economic development, “distributed intelligence” and now is the time for semantic technology investments to strengthen portfolios.

6.1 Usability of the SmartResource Results

The developed methodology of resource adaptation and its prototype implementation can be used by ICT industries in tasks related to the problem of Enterprise Application Integration and to less global problems of e.g. legacy application adaptation. In addition, the proposed solution is compatible with the existing open W3C standard RDF, which provides rich semantic descriptions to resource data and hence enables resource maintenance by future specialized Intelligent Web Services or applications.

To apply the developed approach for the above mentioned tasks, the following steps must be performed:

– Development of the library (centralized or decentralized storage) of reusable programming components according to the decomposition model recommended in GAF. In the development process both large and small software development companies can participate applying their unique expertise in specific component implementations and providing it over

the world for a certain price. The great advantage for the component owners is that, according to GAF, their components are meant to be discovered and linked to the specific adapter automatically thanks to the ontology of the components.

- Ontology of the software components must be engineered to automate the process of their search and acquisition. For this purpose, the existing types of the components must be systematized that requires involvement of a comprehensive player from software market.
- Development of the ontology of templates, which contains a hierarchy of primitives from the canonical forms of different data models (XML, relational database, UML, etc.) and mappings between them. Decomposition of the existing data models into hierarchies of corresponding primitives requires extensive expertise in the field of domain modeling. A large software development company, using the described technology, could provide tools of automated transformation between different data models.

Communication providers can benefit from the available project results in the following way. GAF assumes adaptation and integration of the remote applications, too. This means that to assembly the necessary adapters, the software components, which provide networked connectivity, must be available in the library/ontology of the components. Therefore, software development companies would implement a specific components-connectors bound (configured) to a concrete communication provider, which will be placed in the library and will be further used by adapter purchasers. In the prototype implementation conventional information networked channels are used for connectivity between adapters, e-mail services are used to deliver diagnostic requests to a human expert. The diagnostic request was also tested on a reach/availability from the mobile communicator and the diagnostic interface, too. In the implementation plans there is a SMS-driven human expert notification that is also very beneficial functionally for communication providers.

Another opportunity of the results application is appropriate in the environment of heterogeneous communication operators, where interoperability is required for proper transaction transfer between operators and for integral representation of a state of communication system, or mobile user's state/condition. However, this needs additional analysis of the results in the mentioned context.

6.2 Application Areas

The designed General Adaptation Framework and its implementation due to their original universality are supposed to find applications in various domains, in which distributed heterogeneous resources exist and problems of interoperability and integration into dynamic open environments are emerging.

Besides its main application area (integration of industrial assets), more than once the SmartResource activity results were analyzed in the context of such application areas as Wellness (integration of human patients with embedded medical sensors, doctors-experts and medical web services), Ecology (natural environment with sensors, human experts in an environmental monitoring and Web Services for environmental diagnostics and prediction), Organizational management (staff/students with corresponding monitored organizational data, managers and automated systems for organizational diagnostics and management), Video Security Systems (objects under observation, monitoring experts and video/image automated processing tools), etc.

Expert analysis of recent results and further brainstorming session have revealed their applicability in the Sports domain. Currently, many kinds of human wearable and implanted

sensors exist and their integration could provide a comprehensive data set about a dynamics of a sportsman's state. The SmartResource General Adaptation Framework in this case could be applied for the adaptation of the heterogeneous sensors to a unified environment and their data integrated to a comprehensive semantic data model. Data stored with these assumptions, can be available to sophisticated analytical software (even remote) or human experts, which are also supposed to be integrated to the same medium. As a concrete use case, we can consider e.g. some neuro-fuzzy online predictor, which analyzing a track of ski-jumper's state changes, gives a real-time instruction, e.g. about a right posture. The real-time instruction is very helpful for different sportsmen (swimmers, runners, water-jumpers, bodybuilders and so on) during their training; the corrections of the loading are made depending on the context (human condition, endured traumas, weather conditions, etc.). The output diagnosis of the decision making service can be not so demanding to a response time, like a generation of a monthly, yearly individual training schedules. The latter have been formalized, classified and specified rather well by now that makes them easy to represent in a form of ontology. Sportsmen training and instructing is a very relevant domain for automated learning services, which after the adaptation can learn on the unified sportsmen training data and act as an expert service in future.

Next application area covers various enterprise-wide knowledge management systems, research and development activities management systems, which integrate numerous heterogeneous companies' branches and coordinate their processes, providing an integral and unified representation interface. Enterprise Resources Planning (ERP) is a more concrete application area example: representation of the state of resources through the whole enterprise in the integral view is a current challenge for many large companies today (e.g. integration of reports in Excel, XML and different standard into one). Very often within one big company product or project data, which are distributed among many filial parts in heterogeneous formats/systems, must be transformed to a common format to enable determining the similarities and intersections between the products and projects.

Tender management (evaluation of subcontractors): such companies as Microsoft could utilize the project results for building a management system of the tender activities carried out among numerous heterogeneous 3rd party vendors. For this, the restrictions on specification of the required component/subsystem are formalized in a unified form (according to our solution it will be RDF/RSCDF) to enable automated semantic match with a corresponding descriptions of the 3rd party vendor solutions.

Statistical information gathering e.g. in Automobile Industry is also possible implementation area. Manufacturers could accumulate statistical data integrating sensor/alarm data from embedded blocks inside car systems. Integration of heterogeneous data takes place here and its further analysis would help in planning production strategies.

6.3 Further Development

Semantic Web standards are not yet supporting semantic descriptions of resources with proactive behavior. However as our research shows [11], to enable effective and predictive maintenance of an industrial device in distributed and open environment, it will be necessary to have autonomous agent based monitoring over device state and condition and also a support from remote diagnostics Web-Services. This means that the description of a device as a resource will require also a description of proactive behavior of autonomous condition monitoring applications (agents, services) towards effective and predictive maintenance of the device. For that we are recently developing another extension of RDF, which is Resource Goal/Behavior Description Framework

(RGBDF), to enable explicit specification of maintenance goals and possible actions towards faults monitoring, diagnostics and maintenance. Based on RSCDF and RGBDF and appropriate ontological support, we also plan to design RSCDF/RGBDF platforms for smart resources (devices, Web-services and human experts) equipped by adapters and agents for proactivity, and then to apply several scenarios of communication between the platforms towards learning Web-services based on device data and expert diagnostics to enable automated remote diagnostics of devices by Web-services (see Figure 18).

Another challenge for Semantic Web is the contradiction between the concept of centralized and shared ontology to enable global interoperability and decentralized nature of today's global businesses. Actually the heterogeneity of ontologies is already the fact, which prevents inter-consortia interoperability. Discovering necessary resource or service in the network, which is heterogeneous on ontology level, requires specific solutions, among which semantic peer-to-peer resource discovery and context-sensitive ontologies can be an option. One of the targets for our project is a planned implementation of such condition monitoring, remote diagnostics and predictive maintenance scenarios, which can be managed in decentralized P2P heterogeneous environment. The scenarios assume agent-based interoperation of multiple devices, multiple services and multiple experts, which allows discovery of necessary experts in P2P network, using their experiences to learn remote diagnostics Web-services, making online diagnostics of devices by integrating diagnoses from several services, learning models for a device diagnostics based on online data from several distributed samples of similar device, etc.

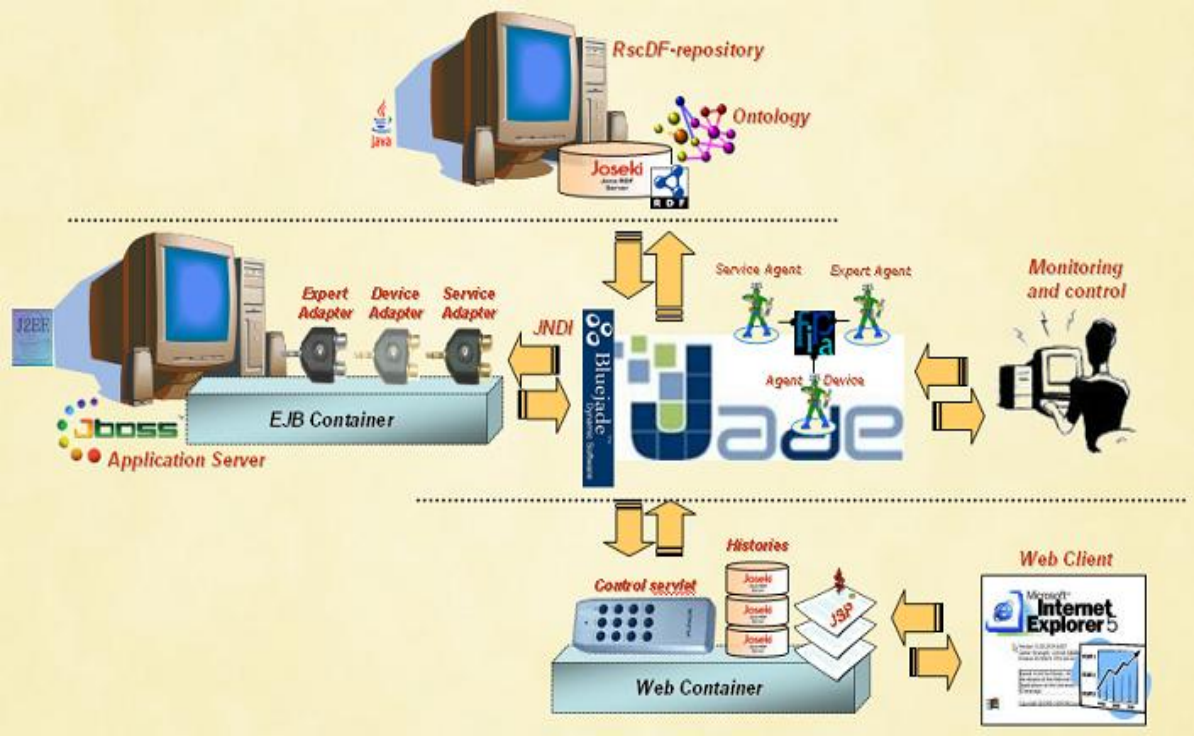

Figure 18 - Preliminary architecture of the SmartResource prototype environment v.2.0

In general, the project's efforts strive to catalyze the evolution of RDF towards two directions: RSCDF (dynamics and context awareness) and RGBDF (proactivity and self-maintenance) and the ultimate result have to be a set of open standards that enable the GUN architecture (Figure 19).

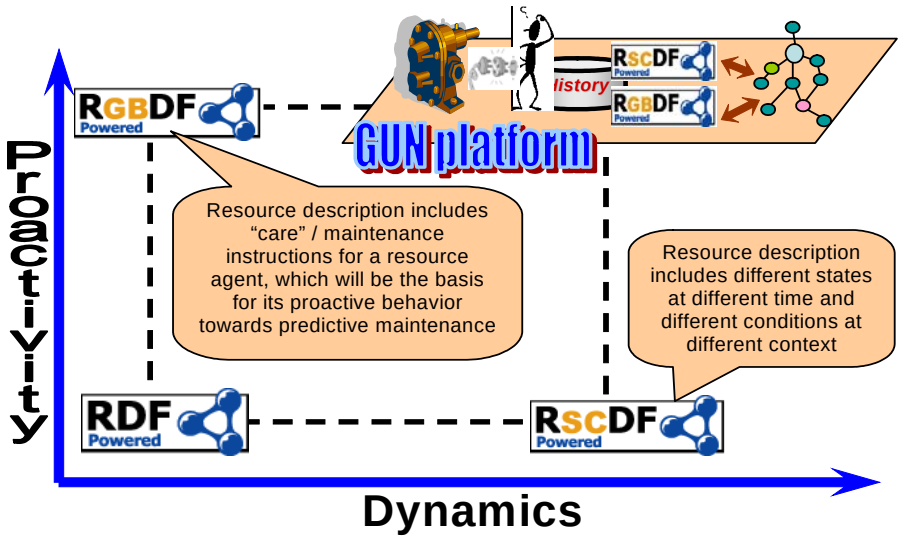

Figure 19 - Evolution of RDF towards GUN platforms through dynamics and proactivity

As it was mentioned above, the GUN environment is meant for online condition monitoring and predictive maintenance of various industrial resources. Utilization of RSCDF and RGBDF allows creation of agent-driven GUN platforms for each industrial resource, where all data related to monitoring, diagnostics and maintenance of the resource will be collected in the resource history ("lifeblog") and managed by the resource agent [35].

Acknowledgement

This research has been performed as a part of the SmartResource ("Proactive Self-Maintained Resources in Semantic Web") project in Agora Center (University of Jyväskylä, Finland) and funded by TEKES and industrial consortium of following companies: Metso Automation, TeliaSonera, TietoEnator and Science Park.

References

1. Adaptive Services Grid (ASG) project, http://asg-platform.org/.
2. Apte A., Java™ Connector Architecture: Building Enterprise Adaptors, Publisher: Sams Publishing, ISBN: 0-672-32310-9, p. 360.
3. BMC Press, Enterprise Application Integration: the Risks Behind the Rewards, White Paper, BMC Software, www.bmc.com, 11th December 2003.
4. Business Process Execution Language for Web Services (BPEL4WS) Version 1.1, May 2003, http://www-128.ibm.com/developerworks/library/ws-bpel/.
5. Cardoso J., Miller J., Su J., Pollock J., Academic and Industrial Research: Do Their Approaches Differ in Adding Semantics to Web Services? In: Proc. of First International Workshop on Semantic Web Services and Web Processes Composition (SWSWPC 2004), Cardoso, J., Sheth, A. (Eds.), LNCS, Springer-Verlag Heidelberg , Vol. 3387, pp.14-21. 2005. ISBN: 3-540-24328-3.
6. Context-aware Business Application Service Co-ordination in Mobile Computing Environments – CASCOM project, http://www.ist-cascom.org/.

7. Davis M., The Business Value of Semantic Technology, In: Proceedings of the Second Annual Semantic Technologies for E-Government Conference, September 2004, McLean, Virginia, USA.

8. Ermolayev V., Keberle N., Plaksin S., Kononenko O., Terziyan V., Towards a Framework for Agent-Enabled Semantic Web Service Composition, International Journal of Web Service Research, Idea Group, ISSN: 1545-7362, Vol. 1, No. 3, pp. 63-87.

9. Farrar, S., Lewis, W., and Langendoen, T. A Common Ontology for Linguistic Concepts. In Proceedings of the Knowledge Technologies Conference, Seattle, Washington, March 10-13, 2002.

10. Jess – Rule Engine for the Java Platform, official website at http://herzberg.ca.sandia.gov/jess/, last accessed 10th Apr 2005.

11. Kaikova H., Khriyenko O., Kononenko O., Terziyan V., Zharko A., Proactive Self-Maintained Resources in Semantic Web, Eastern-European Journal of Enterprise Technologies, Vol. 2, No. 1, ISSN: 1729-3774, Kharkov, Ukraine, pp. 4-16.

12. Kaykova O., Khriyenko O., Kovalainen M., Zharko A., Visual Interface for Adaptation of Data Sources to Semantic Web, In: Proceedings of the IASTED International Conference on Software Engineering (SE 2004), February 17-19, 2004, Innsbruck, Austria, ACTA Press, ISBN: 0-88986-381-4, ISSN: 1027-2666, pp. 544-547.

13. Kaykova O., Khriyenko O., Naumenko A., Terziyan V., Zharko A., RSCDF: A Dynamic and Context-Sensitive Metadata Description Framework for Industrial Resources, In: *Eastern-European Journal of Enterprise Technologies*, Vol. 3, No. 3, June 2005, ISSN: 1729-3774 (to appear).

14. Khanna R., Top Challenges in Integration Projects, *White Paper*, ITtoolbox EAI – Enterprise Application Integration section of ITtoolbox - online information provider on ICT, www.ittoolbox.com, 21st May 2004.

15. Khriyenko O., Terziyan V., Context Description Framework for the Semantic Web, *Working Draft*, 30 March 2005 16 pp., http://www.cs.jyu.fi/ai/papers/JBCS-2005.pdf.

16. Khriyenko O., Terziyan V., OntoSmartResource: An Industrial Resource Generation in Semantic Web, In: R. Schoop, A. Colombo, R. Bernhardt, and G. Schreck (eds.), Proceedings of the 2nd IEEE International Conference on Industrial Informatics (INDIN'04), Berlin, Germany, 24-26 June, 2004, IEEE, ISBN: 0-7803-8513-6, pp. 175-179.

17. Lara R., Roman D., Polleres A., and Fensel D. A Conceptual Comparison of WSMO and OWL-S. In Proc. of *European Conference on Web Services*, Erfurt, Germany, Sept. 27-30, 2004, pp. 254-269.

18. Lucena V., Facet-Based Classification Scheme for Industrial Automation Software Components, Proc. of Sixth International Workshop on Component-Oriented Programming (WCOP 2001), Budapest, Hungary, 2001.

19. Malucelli A. and Oliveira E., Ontology-Services to Facilitate Agents' Interoperability, In PRIMA'03 - 6th Pacific Rim International Workshop On Multi-Agents, LNAI, Springer-Verlag. November 7-8, Seoul, Korea, 2003.

20. METEOR-S: Semantic Web Services and Processes, LSDIS Lab, University of Georgia, http://lsdis.cs.uga.edu/ and http://swp.semanticweb.org/.

21. Metso Automation's customer magazine, (2003) Automation, 1, 7-9.

22. Miller E., Sperberg-McQueen C.M., On mapping from colloquial XML to RDF using XSLT, Proc. of W3C Extreme Markup Languages 2004, August 3, 2004; Montreal, CA.

23. Nierstrasz O., Dami L., Component-Oriented Software Technology, Object-Oriented Software Composition, Nierstrasz O. and Tsichritzis D. (Eds.), pp. 3-28, Prentice Hall, 1995.

24. Nierstrasz O., Gibbs S., Tsichritzis D., Component-Oriented Software Development, Communications of the ACM, Vol. 35, No. 9, Sept. 1992.

25. Nikitin S., Terziyan V., Tsaruk Y., Zharko A., Querying Dynamic and Context-Sensitive Metadata in Semantic Web, In: V. Gorodetsky, J. Liu, and V.A. Skormin (Eds.): *Proceedings of the International Workshop on Autonomous Intelligent Systems – Agents and Data Mining (AIS-ADM-2005)*, Springer, LNAI 3505, 2005, pp. 200-214.

26. Ontology Web Language for Web Services (OWL-S) 1.0, http://www.daml.org/services/owl-s/1.0/.

27. Paolucci M., Srinivasan N., and Sycara K. Expressing WSMO Mediators in OWL-S. *International Semantic Web Conference 2004*, W6: 120–134, 2004. Workshop notes VI: Semantic Web Services.

28. Semantic Web Enabled Web Services (SWWS) project, http://swws.semanticweb.org/.

29. Semantic Web Services Initiative/Semantic Web Services Architecture (SWSI/SWSA), http://www.daml.org/services/swsa/.

30. Simple Object Access Protocol (SOAP) 1.1, http://www.w3.org/TR/soap/.

31. Sivashanmugam K., Miller J., Sheth A., and Verma K.**,** Framework for Semantic Web Process Composition, International Journal of Electronic Commerce, Winter 2004-5, Vol. 9(2) pp. 71-106.

32. Sun Press, J2EE™ Connector Architecture Specification, v. 1.5, Sun Microsystems Inc., www.sun.com, 24th November 2003.

33. Terziyan V., Semantic Web Services for Smart Devices in a "Global Understanding Environment", In: R. Meersman and Z. Tari (eds.), On the Move to Meaningful Internet Systems 2003, LNCS, Vol. 2889, Springer-Verlag, 2003, pp.279-291.

34. Terziyan V., Semantic Web Services for Smart Devices Based on Mobile Agents, In: *International Journal of Intelligent Information Technologies*, Vol. 1, No. 2, April-June 2005, Idea Group, ISSN 1548-3657, pp. 43-55.

35. Terziyan V., SmartResource: Utilizing Semantic Web Services to Monitor Industrial Devices, In: Proceedings of the XML Finland 2005: XML - the Enabling Technology for Integrating Business Processes, March 8-9, 2005, Pori, Finland.

36. Terziyan V., Kononenko O., Semantic Web Enabled Web Services: State-of-Art and Industrial Challenges, In: M. Jeckle and L.-J. Zhang (eds.), Web Services - ICWS-Europe 2003, Lecture Notes in Computer Science, Vol. 2853, Springer-Verlag, 2003, pp. 183-197.

37. Terziyan V., Zharko A., Semantic Web and Peer-to-Peer: Integration and Interoperability in Industry, In: International Journal of Computers, Systems and Signals, IAAMSAD, ISSN 1608-5655, Vol. 4, No. 2, pp. 33-46.

38. Teschke T., Ritter J., Towards a Foundation of Component-Oriented Software Reference Models, In: Greg Butler, Stan Jarzabek (Hrsg.): Generative and Component-Based Software Engineering, Lecture Notes in Computer Science (LNCS), 2177, Springer Verlag, S.70-84, 2001.

39. UDDI Spec Technical Committee Specification, 19 July 2002, http://www.uddi.org/specification.html.

40. Web Service Description Language (WSDL) 1.1, http://www.w3.org/TR/wsdl.

41. Web Services Modeling Ontology (WSMO), http://www.wsmo.org/.

42. XR's homepage: XML-to-RDF transformation format, http://w3future.com/xr/, last accessed 19th Oct 2004.