TEKES

SmartResource Project
(2004-2007)

Final Project Report

"SmartResource - Proactive Self-Maintained Resources in Semantic Web: Lessons Learned"

Industrial Ontologies Group

Agora Center, University of Jyväskylä

JSP Jyväskylä Science Park · metso automation · University of Jyväskylä · TeliaSonera · TietoEnator Building the Information Society · ABB

March, 2007
Jyväskylä, Finland

SmartResource Tekes Project – Final Project Report (2007)

Main Contact Persons

Head of SmartResource Industrial Consortium (Steering Committee Head) **Dr. Jouni Pyötsiä**, Metso Automation Oy.

Jouni.Pyotsia@metso.com , Tel.: 040-548-3544

SmartResource Contact Person **Prof. Timo Tiihonen**, Vice-Rector, University of Jyväskylä

tiihonen@it.jyu.fi , Tel.: 014-260-2741

SmartResource Project Leader **Prof. Vagan Terziyan**, Agora Center, University of Jyväskylä

vagan@it.jyu.fi , Tel.: 014-260-4618

Tekes Contact Person, technology expert **Jari Räihä**, TEKES.

Jari.räihä@tekes.fi , Tel.: 010 60 24184
GSM: 040 521 2126

Tekes Contact Person, technology expert **Timo Taskinen**, TEKES.

Timo.taskinen@tekes.fi , Tel.: 010 60 24846
GSM: 0400 648 504

Project's webpage:

http://www.cs.jyu.fi/ai/OntoGroup/SmartResource_details.htm

Metso Automation

Metso Automation specializes in automation and information management application networks and systems, field control technology and life cycle performance services. Its main customers are the pulp and paper as well as power, energy and oil and gas industries.

(2004 – 2007)

TeliaSonera

TeliaSonera is the Nordic & Baltic telecommunications leader. TeliaSonera offers reliable, innovative and user-friendly services for transferring and packaging of voice, images, data, information, transactions and entertainment. TeliaSonera also provides wholesale carrier services between selected destinations in Europe and across the Atlantic. Depending on the market position, TeliaSonera offers a complete service portfolio or a focused range of services.

(2004 – 2007)

TietoEnator

TietoEnator is consulting, developing and hosting its customers' digital businesses. TietoEnator is among the leading architects in building a more efficient information society. With over 15 000 experts, it is one of the largest IT services providers in Europe. Leading-edge know-how is geared towards developing innovative IT solutions that realize and digitalize the visions of the customers. Company has chosen to focus on areas where it has the deepest industry expertise. The principal ones globally are banking, telecom, healthcare and forest.

(2004 – 2005)

ABB

ABB is a global leader in power and automation technologies that enable utility and industry customers to improve performance while lowering environmental impact. With their technology leadership, global presence, application knowledge and local expertise, company offers products, systems, solutions and services that allow its customers to improve their operations – whether they need to increase the reliability of a power grid or raise productivity in a factory.

(2006 – 2007)

Jyväskylä Science Park

Jyväskylä Science Park aims to reinforce technology-based business and its preconditions. Their goal is to make Jyväskylä an internationally successful, people-based center of technology that utilizes the region's multibranched industrial base. By offering a broad service package, strong networks and competitive expertise, company creates sector-based information and know-how for its clients, a framework for developing their businesses, and a purpose-built operating environment. They help clients refine good ideas into successful business and strengthen the region's sector-based expertise.

(2004)

University of Jyväskylä, Agora Center

Industrial Ontologies Group

Vagan Terziyan
Prof., Ph.D., Project Leader

vagan@it.jyu.fi
http://www.cs.jyu.fi/ai/vagan/

2003-2007

Olena Kaykova
Ph.D., Senior Researcher
helen@it.jyu.fi
http://www.cs.jyu.fi/ai/helen

2003-2007

Oleksiy Khriyenko
M.Sc., Researcher & Developer
olkhriye@cc.jyu.fi
http://www.cc.jyu.fi/~olkhriye/

2003-2007

Sergiy Nikitin
M.Sc., Researcher & Developer
senikiti@cc.jyu.fi
http://people.jyu.fi/~senikiti/

2004-2007

Anton Naumenko
M.Sc., Principal developer
annaumen@cc.jyu.fi
http://www.cc.jyu.fi/~annaumen/

2004-2007

Artem Katasonov
PhD, Researcher & Developer
artem.katasonov@jyu.fi

2006-2007

Andriy Zharko
M.Sc., Researcher & Developer
anzharko@cc.jyu.fi
http://www.cc.jyu.fi/~anzharko/

2003-2005

Yaroslav Tsaruk
M.Sc., Researcher & Developer
yatsaruk@cc.jyu.fi
http://people.jyu.fi/~yatsaruk/

2004,2006

Dmytro Kovtun
M.Sc., Researcher & Developer
dmkovtun@cc.jyu.fi
http://people.jyu.fi/~dmkovtun/

2004

Jari Marttinen
B.Sc., Trainee
jarmart@cc.jyu.fi

2004

Oleksandr Kononenko
M.Sc., Researcher & Developer
oleksandr.kononenko@nokia.com

2003-2004

http://www.cs.jyu.fi/ai/OntoGroup/

SmartResource

Proactive Self-Maintained Resources in Semantic Web: Lessons Learned

SmartResource - Proactive Self-Maintained Resources in Semantic Web: Lessons Learned

Vagan Terziyan (editor)[1]

Industrial Ontologies Group, MIT Department, University of Jyvaskyla, P.O. Box 35, FIN 40014 Jyvaskyla, FINLAND
e-mail: vagan@it.jyu.fi

Abstract

The *SmartResource* project (2004-2006) funded by Tekes and industrial companies (Metso Automation, TeliaSonera, TietoEnator, ABB) is officially ending in April 2007. Its objectives were research and development of the large-scale environment for integration of industrial smart devices, web services and human experts based on Semantic Web and agent technologies. A prototype platform for self-maintained smart industrial resources has been designed and implemented for particular tasks of industrial partners. In this paper we will present the summary of research results obtained during the project period and its self-evaluation study. Several lessons have been learned during the project in addition to published results, which we are going to share with scientific community.

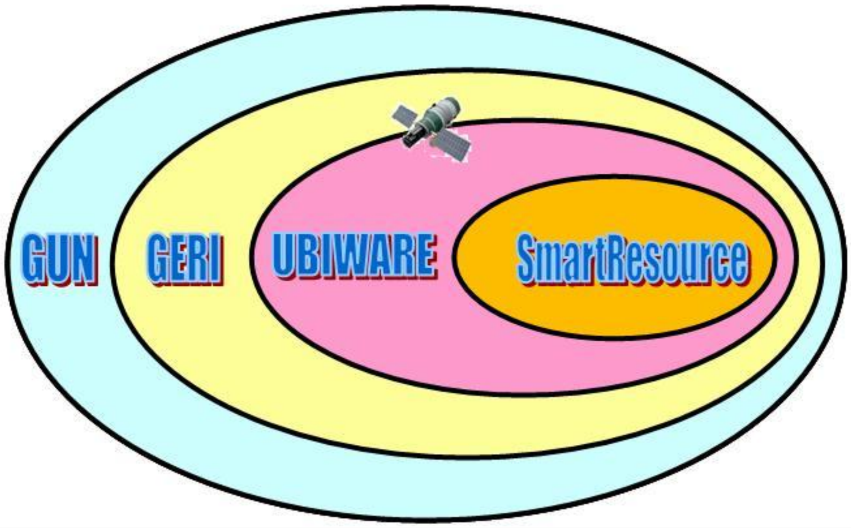
1. Introduction

The *SmartResource* project[2] (2004-2006) funded by Tekes[3] and industrial companies (Metso Automation[4], TeliaSonera[5], TietoEnator[6], ABB[7]) is officially ending in April 2007. Its objectives were research and development of the large-scale environment for integration of industrial smart devices, web services and human experts based on Semantic Web and agent technologies. A prototype platform for e-maintenance has been designed and implemented for particular tasks of industrial partners.

The project belongs to the Industrial Ontologies Group[8] research roadmap (see Figure 1) towards the *Global Understanding Environment (GUN)* (Terziyan, 2003, 2005; Kaykova *et al.*, 2005a). When applying Semantic Web in the domain of ubiquitous computing, it should be obvious that Semantic Web has to be able to describe resources not only as passive functional or non-functional entities, but also to describe their behavior (proactivity, communication, and coordination). In this sense, the word "global" in GUN has a double meaning. First, it implies that resources are able to communicate and cooperate globally, i.e. across the whole organization and beyond. Second, it implies a "global understanding". This means that a resource A can understand all of (1) the properties and the state of a resource B, (2) the potential and actual behaviors of B, and (3) the business processes in which A and B, and maybe other resources, are jointly involved.

---

1 Based on publications of SmartResource project team
2 Web pages of SmartResource project: http://www.cs.jyu.fi/ai/OntoGroup/SmartResource_details.htm
3 Finnish Funding Agency for Technology and Innovation: http://www.tekes.fi/eng/
4 Web site of Metso Automation: http://www.metsoautomation.com/
5 Web site of TeliaSonera: http://www.teliasonera.fi/
6 Web site of TietoEnator: http://www.tietoenator.fi/
7 Web site of ABB: http://www.abb.fi/
8 Web pages of Industrial Ontologies Group: http://www.cs.jyu.fi/ai/OntoGroup/index.html



GUN (Global Understanding Environment) – Proactive Self-Managed Semantic Web of Things - general concept and final destination

GERI (Global Enterprise Resource Integration) – GUN subset related to industrial domains

UBIWARE – middleware for GERI

SmartResource – semantic technology, pilot tools and standards for UBIWARE

Figure 1 - The research roadmap towards GUN.

GUN aims at making heterogeneous resources (physical, digital, and humans) web-accessible, proactive and cooperative. Three fundamentals of such platform are *Interoperability*, *Automation* and *Integration*. Interoperability in GUN requires utilization of Semantic Web standards, RDF-based metadata and ontologies and semantic adapters for the resources. Automation in GUN requires proactivity of resources based on applying the agent technologies. Integration in GUN requires ontology-based business process modeling and integration and multi-agent technologies for coordination of business processes over resources.

Main layers of GUN can be seen in Figure 2. 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 rich 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. Implementation 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.

Figure 2 - The concept of Global Understanding Environment.

When applying the GUN vision, each traditional system component becomes an agent-driven "smart resource", i.e. proactive and self-managing. This can also be recursive. For example, an interface of a system component can become a smart resource itself, i.e. it can have its own responsible agent, semantically adapted sensors and actuators, history, commitments with other resources, and self-monitoring, self-diagnostics and self-maintenance activities. This could guarantee high level of dynamism and flexibility of the interface. Such approach definitely has certain advantages when compared to other software technologies, which are integral parts of it, e.g. OOSE, SOA, Component-based SE, Agent-based SE, and Semantic SE. This approach is also applicable to various conceptual domain models. For example, a domain ontology can be considered as a smart resource, what would allow having multiple ontologies in the designed system and would enable their interoperability, on-the-fly mapping and maintenance, due to communication between corresponding agents.

The roadmap of SmartResource research and development also assumes complex interactions between multiple distributed agents – representatives of Web Services, human experts and smart devices. This necessity is caused by advanced features of the SmartResource environment, e.g.:

- One industrial smart device exploits several Web Services (composition) for more precise diagnostics;
- One Web Service provides diagnostics for multiple smart devices: knowledge reuse cases;
- Learning Web Service utilizes multiple sources of training sets (many smart devices);
- One smart device provides training sets from its history to multiple Learning Web Services;
- Exchange of underlying diagnostic models between Web Services;
- One smart device requests opinions about its conditions and diagnoses from multiple human experts to increase quality of decisions;
- One expert provides diagnostic service to multiple devices (efficient reuse of knowledge).

Additional important aspect that has been studied is trust. This aspect inevitably arises in environments where multiple customers interact remotely with service providers, whose authenticity, credibility and reliability have to be verified. In general, two strategies regarding trust have to be analyzed:

- Exchange of opinions about authenticity and quality of Web Services (QoS) and human experts between smart devices (controlled by their companies-owners);
- Certification of the Web Services and experts by respectable and trusted authorities.

The SmartResource project, in its research and development efforts, has made some steps towards Global Understanding Environment by decomposing it into three main parts, and analyzing each.

The first is the *General Adaptation Framework (GAF)* for semantic interoperability. GAF provides means for semantic description of industrial resources, including dynamic and context-sensitive information. The central part is GAF is played by the Resource State/Condition Description Framework (RscDF). An implementation of GAF for a specific domain is supposed to include also an appropriate RscDF-based domain ontology, an appropriate RscDF Engine and the family of so called "Semantic Adapters for Resource" to provide an opportunity to transform data from a variety of possible resource data representation standards and formats to RscDF and back. For more details about RscDF and GAF see (Kaykova *et al.*, 2005b) and (Kaykova *et al.*, 2005a).

The second is the *General Proactivity Framework (GPF)* for automation and proactivity. GPF provides means for semantic description of individual behaviors by defining the Resource Goal/Behavior

Description Framework (RgbDF). An implementation of GPF is supposed to include also an appropriate RgbDF-based domain ontology, an appropriate RgbDF engine and a family of "Semantic Adapters for Behavior" to provide an opportunity to transform data from a variety of possible behavior representation standards and formats to RgbDF and back. See more on RgbDF in (Kaykova *et al.*, 2005c).

The third is the *General Networking Framework (GNF)* for coordination and integration. GNF provides means for description of a group behavior within a business process. It specifies the Resource Process/Integration Description Framework (RpiDF), and an implementation of GNF is supposed to include also an appropriate RpiDF-based domain ontology, an appropriate RpiDF engine and a family of "Semantic Adapters for Business Process" to provide opportunity to transform data from a variety of business process representation standards and formats to RpiDF and back.

Finally, GUN ontologies will include various available models for describing all GAF-, GPF- and GNF- related domains. The basis for interoperability among RscDF, RgbDF and RpiDF is a universal triplet-based model provided by RDF and two additional properties of a triplet *(true_in_context* and *false_in_context*). See more about contextual extension of RDF in (Khriyenko and Terziyan, 2006).

2. Smart Resources: what are they?

In SmartResource project we considered resources to be smart if they are Web-accessible, proactive, self descriptive and self managed. What are "resources" however? See Figure 3. Traditionally Semantic Web community considers traditional Web-resources (documents, software, databases, services, etc) as a subject of semantic enhancement (RDF annotation driven by shared ontology and making the resource self-descriptive). However we consider such consideration as essentially restricted. Industrial domains consist of quite a lot of other categories of resources (machines, humans, processes, etc) and interoperability requirements should be valid also to these categories.

In our approach we consider the following categories of resources as the subject for "smartening" them (i.e. connecting them to the Web and making them proactive, autonomous and self-descriptive):

- *Software*: Software and software components, operation systems, tools, Web-services, etc.;
- *Data:* Electronic documents, warehouses, databases, histories, diaries, lifeblogs, digital media resources, etc.
- *Devices:* all kind of devices, machines, sensors, actuators, adapters, communicators, switches, routers, etc. and their components;
- *Humans:* Users, customers, service providers, buyers, sellers, workers, operators, experts, etc.;
- *Communication systems and networks:* PANs, LANs, MANs, WANs, RFIDs, WiFi, WiMax, LTE, etc.
- *Organizations:* various compositions of various resources selected and integrated for certain purpose, e.g. companies, universities, networks, etc.;
- *Processes:* natural, controlled or goal-driven, dynamics of organizations;
- *Concepts, Models and Ontologies:* various concepts, models and ontologies, which formally describe various resources, their organizational structure, dynamics and coordination;
- *Messages:* all kind of messages various resources exchange among themselves during their lifecycle;
- *Standards:* all kind of standards according to which resources are produced, described, used, operate, communicate etc;
- and etc.

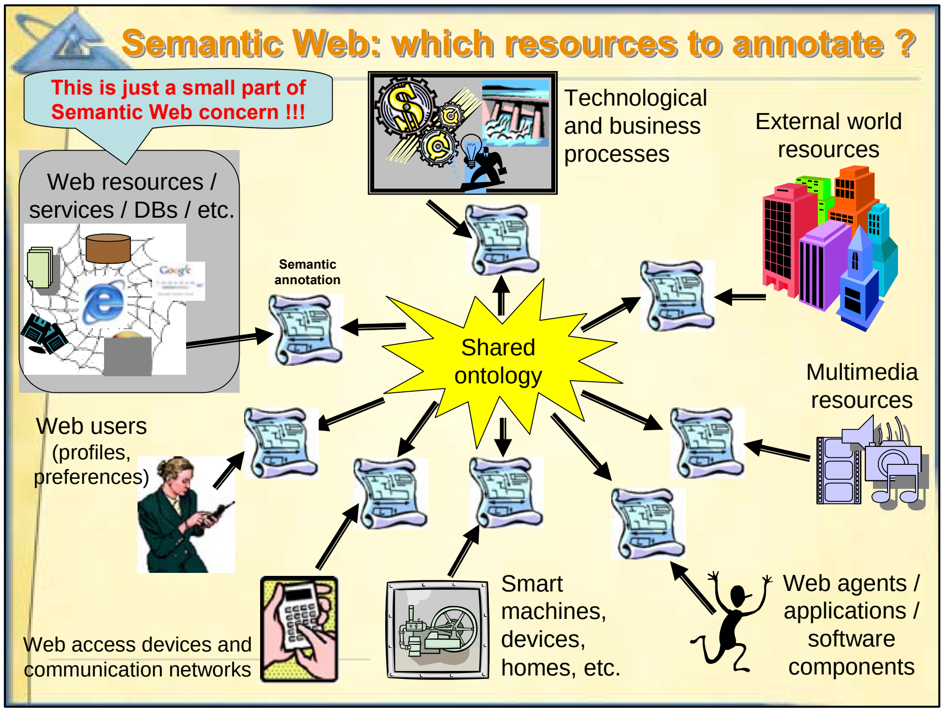


Figure 3 – Extended view to Semantic Web resources

The shift from the Web of documents and software to the Web of Things should affect the Semantic Web research roadmap. So far, the concepts of (semantic) discovery, selection, composition, orchestration, integration, invocation, execution monitoring, coordination, communication, negotiation, context awareness, etc. were mainly applied to the Web-services domain. In future, however, all these concepts should be modified to be applicable also to a resource from the Web of Things. Also, some new things should be taken into account, such as e.g. (semantic) diagnostics, forecasting, control, maintenance, learning, etc. (see Figure 4).

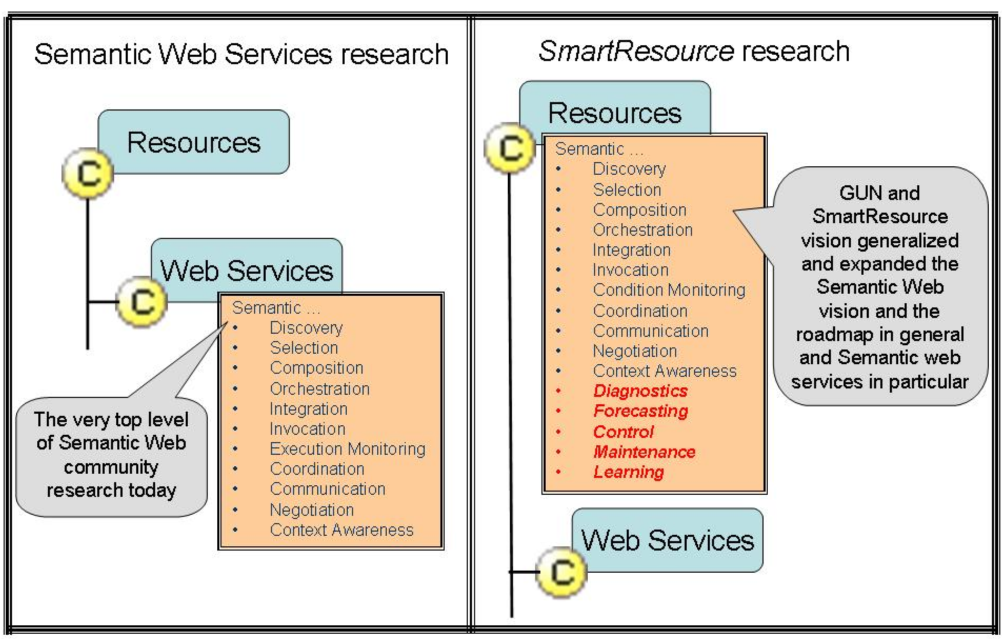


Figure 4 - Shifting Semantic Web roadmap to the World of Things domain

3. General Adaptation Framework

One of the most important challenges of GUN in general and SmartResource in particular is to provide opportunity to design semantic adapters for heterogeneous resources with as minimal effort as possible and with maximal reuse of previously designed adapters and their component when designing new ones (see Figure 5). Ideally the adapter should be that kind of software that is able to automatically reconfigure itself for each new resource based on its declarative description. As a result of adaptation any parameters observed, measured or collected elsewhere about the resource will be available in the same semantically rich format (RDF-based) referring some shared ontology. We developed RscDF (Resource State/Condition Description Framework) as an appropriate format for adapters output (Kaykova et al. (2005b)). RscDF extends RDF by making it more suitable for semantic annotation of dynamic and context-sensitive data about the resources.

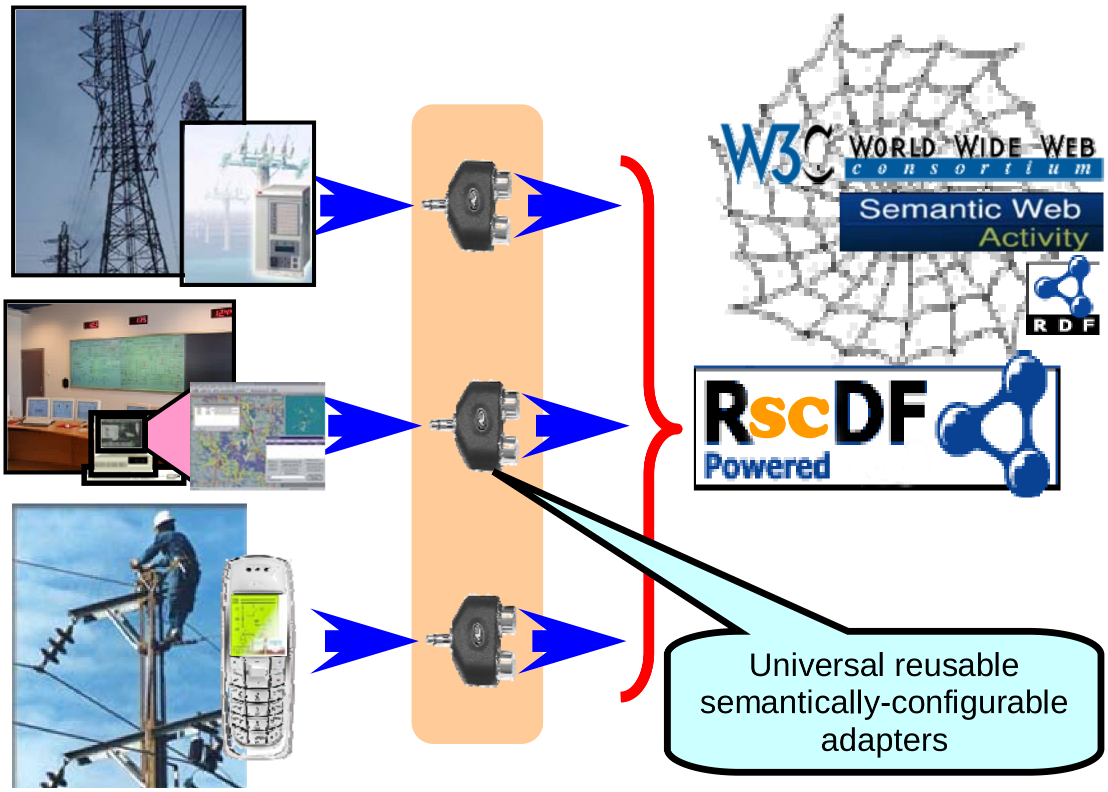


Figure 5 - The challenge of universal self-configurable adapters.

It is really challenging task to adapt extremely heterogeneous real world resources to Web environment. The task must be solved by creating set of reusable (hardware and software) components for the adapters (see Figure 6) and a smart way how to automatically design an adapter for some resource by combining existing components based on the resource semantic description. In (Kaykova *et al.*, 2005a) a General Adaptation Framework has been discussed to target the problem. Resource Adapters based on General Adaptation Framework are supposed:

- to enable to connect industrial resources to GUN Environment;
- to add semantics to the resource data;

- to encode data into RscDF, which enables semantic description of dynamic and context-sensitive resources;
- to be build from hardware, software and even "human" components;

General Adaptation Framework provides tools and technology for semi-automated creation of adapters from reusable components and templates based on Semantic Technology.

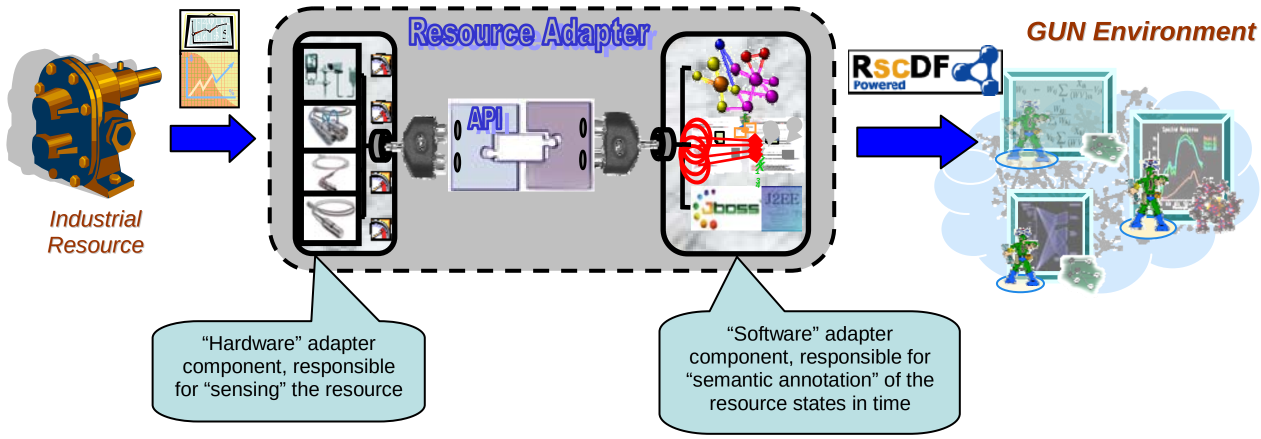


Figure 6 - Architecture of an adapter linking any industrial resource to GUN environment

One of obstacles 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 insufficient 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 motivated one of objectives of the SmartResource project (Adaptation Stage) - a design of General Adaptation Framework, which provides a methodology for designing adapters to semantically enrich and transfer data from various formats to RscDF (dynamic and context-sensitive RDF) and back. In the project the approach was tested on concrete implementations - adapters for three different samples of heterogeneous resources (device data, expert interface, Web-Service).

3.1. Pilot implementation of General Adaptation Framework

In the project we distinguished two aspects of adaptation: data model transformation and Application Programming Interface (API) adaptation. 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 with it. The strength of the two-stage transformation is in reuse of a variety of existing powerful tools for data model mapping 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 a 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 two 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, classified and arranged into a centralized/decentralized library. Such components, in a vision of GAF, are building blocks for automated assembly of a concrete adapter on the fly. The automated component integration is performed using ontology of components and the resulting adapter is run as an EJB (Enterprise Java Bean) component on a JBoss Application server in our implementation (see Figure 7).

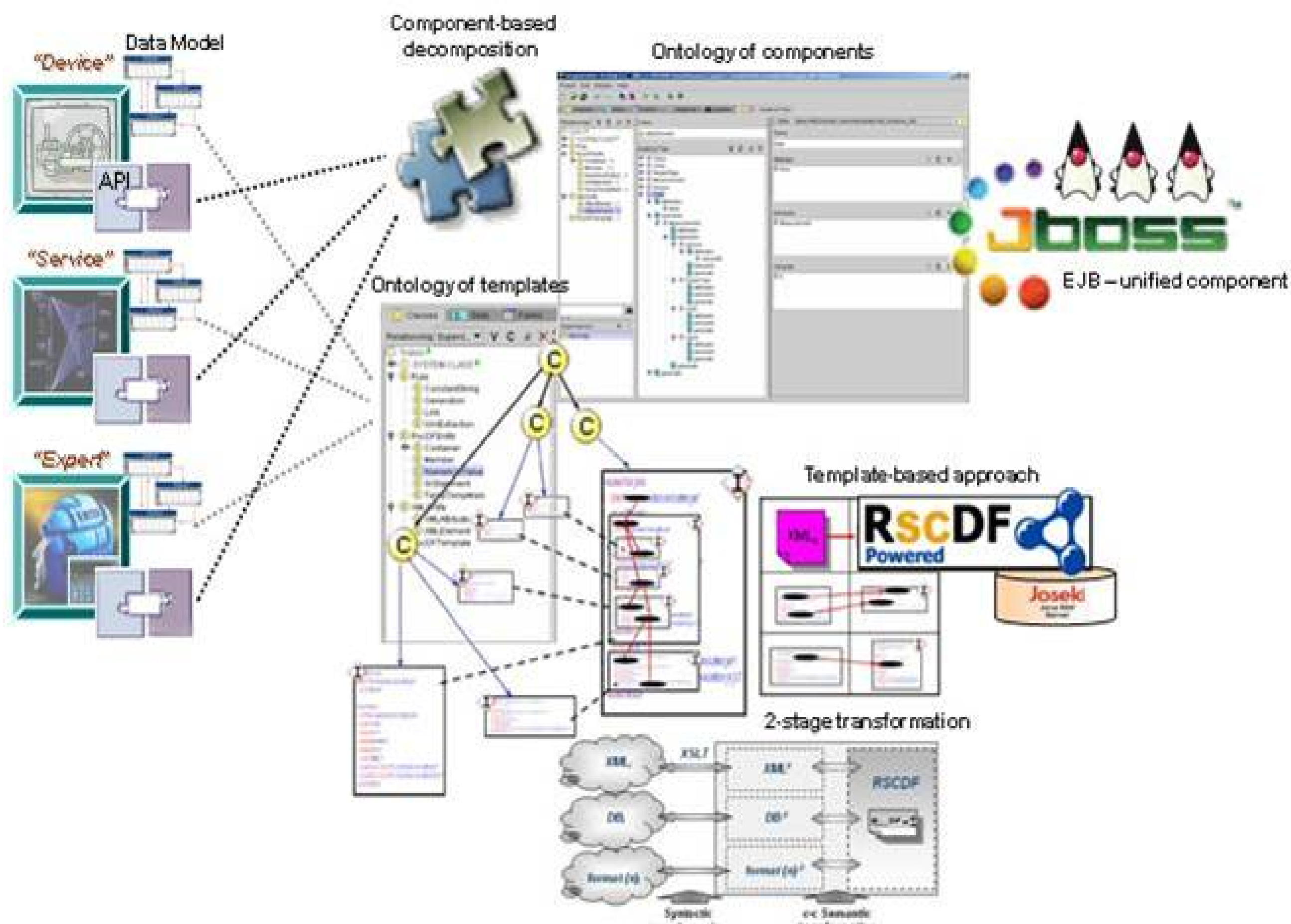


Figure 7 – General Adaptation Framework architecture

To 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.

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 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 (RscDF), 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.

3.2. Where to utilise General Adaptation Framework

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.

The designed General Adaptation Framework and its implementation 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 Telemedicine and 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.

4. General Proactivity Framework

Another important challenge of GUN in general and SmartResource in particular is to make every domain resource proactive, which means able to autonomously behave towards achieving certain goals depending on its role in the domain. Such resources should be able to initiate own self-diagnostics and self-maintenance or outsource diagnostic and maintenance tasks from other resources. Sure that such behavior depends on the nature and the type of the resource, its placement in the environment, relations with other resources, environmental parameters, etc. In SmartResource project we implemented autonomy and proactivity of resources by means of software agents. The main challenge however was to avoid designing different agents for each of heterogeneous resources but implement just one universal agent (like an artist), which will be able to play any declaratively described behavior according to its current role. We require designing such reusable declarative behavior descriptions to be made with as minimal effort as possible and with maximal reuse of previously designed behaviors and their components when designing new ones (see Figure 8). Ideally the agent should be that kind of software that is able to automatically reconfigure itself for each new resource based on declarative description of this resource role in the domain or within some business process. In SmartResource project we designed RgbDF (Resource Goal/Behavior Description Framework) as a tool for semantic annotation of behavioral properties of the resource (goals, plans, roles, actions, intensions, etc.). RgbDF extends RDF by making it more suitable for semantic annotation of data about proactive and autonomous behavior of the resources (Kaykova *et al.* (2005c)).

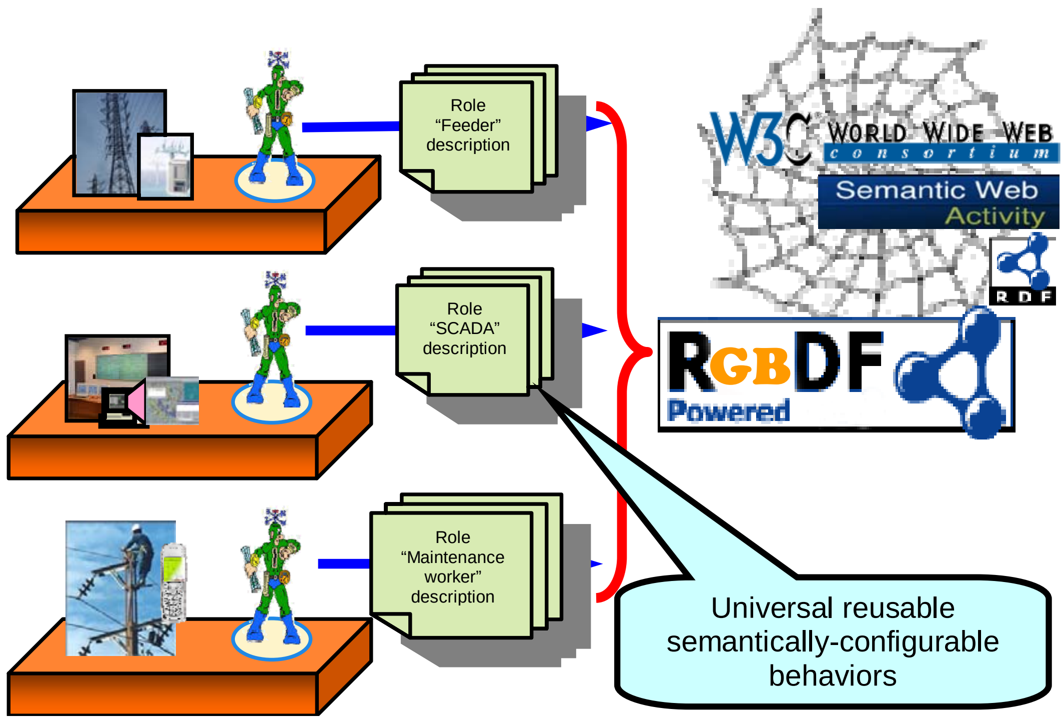


Figure 8 - The challenge of universal self-configurable behaviors

Consider example scenario in Figure 9. It consists of three different resources (some industrial device, some Web-service for intelligent diagnostics and some human expert). The basic idea of this scenario is that the device is self-monitoring itself and in case of some fault or alarm initiates request to the expert for human diagnostics or to the Web-service for automated diagnostics. Let Web-service be some intelligent tool, which is based on Neural Network diagnostics and it should be learned on some training set of already diagnosed samples prior to making diagnostics itself. Consider human expert as the best but expensive source if diagnostic decisions based on data from device sensors. We can split the whole scenario to three scenes. During Scene 1, the agent responsible for the device plays the role “patient”, which means that it monitors own parameters via some sensors and in case of any alarm sends these parameters to human expert for the diagnostics. In this scene the agent of an expert plays the role “diagnostic expert”, which is responsible to reply by naming concrete diagnosis based on requests from the device agent. The agent of Web service in this scene is passive. During Scene 2 of the scenario after device agent have collected enough cases of own diagnoses it can change the role from “patient” to “teacher”, which means that it can provide training samples to the Web-service. Accordingly the agent of the web-service is taking role of “student”, i.e. the one who will learn based on sample set and produces some neural network for future diagnostics. Agent of expert will not play any active role anymore. During Scene 3, after Web-service has learned and is able to make diagnostics automatically, its agent is taking the role “diagnostic expert” and the agent of the device can take the role “patient” back because now it can address all its diagnostic requests to the Web-Service.

The above scenario shows that the roles (i.e. appropriate behaviors) of agents can be chosen and changed depending on current context of the situation, and this means that each agent should be able to download from some shared place the description of a new role whenever needed.

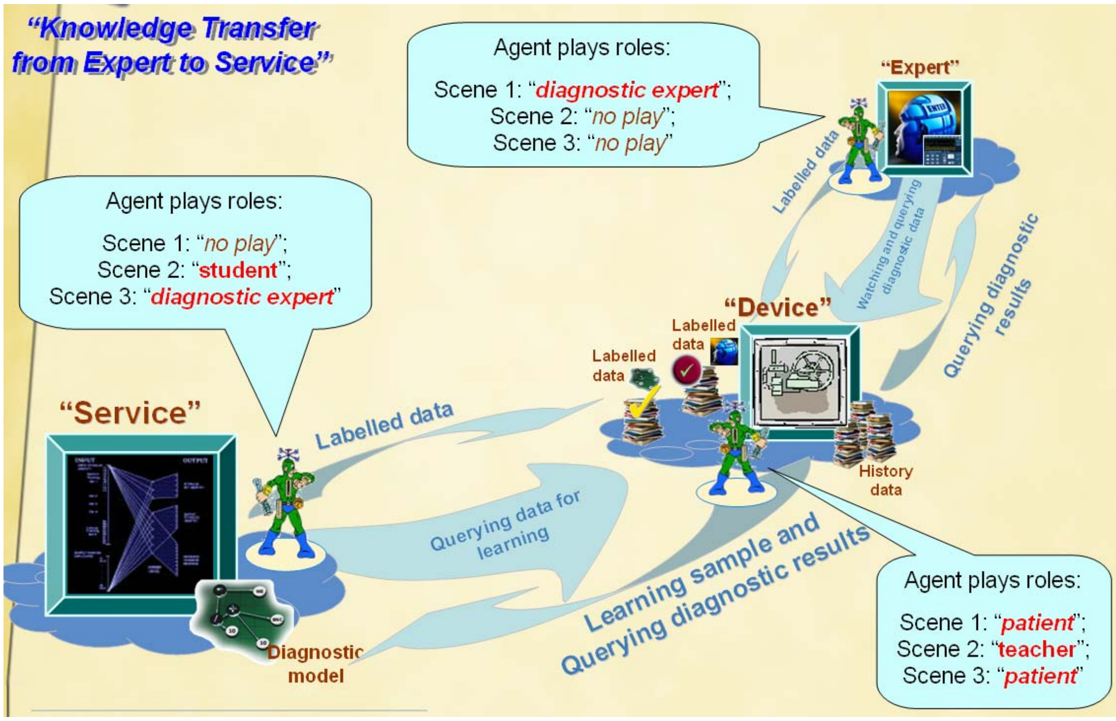


Figure 9 – Sample scenario in which agents can change own roles depending on the context

Autonomous systems must be automatic and, in addition, they must have a capacity to form and adapt their behavior while operating in the environment. Thus traditional AI systems and most robots are automatic but not autonomous - they are not fully independent from the control provided by their designers. Autonomous systems are independent and are able to perform self-control.

In an Agent Environment (as well as in the real world) the base for any interaction is behavior of each individual. Further, integration of these individual behaviors may form behavior of a Multiagent System. A goal-driven behavior means performing a set of rules, which are aimed to achievement of certain goal. In return, goal is a fact which does not exist in a description of the environment, and an agent aims at appearance of the fact. As a result, we have a trio: *behavior* which is driven by certain *goal* and which lies in performing appropriate *actions* following a set of behavioral rules. Each agent should have initial set of those trios (regulated by initial *role*). These trios represent expertise and experience of an agent. As well as in real world agents can exchange their expertise (rules for execution of actions depending on the goals and direct software modules for execution of actions). Availability of a wide spectrum of the trios gives a possibility for agent to automatically divide goals to sub-goals and to create a chain of nested trios. An agent role means aggregating goals corresponding to a specific purpose of the agent. Individual role does not assume a fixed set of activities, the set of the goals can be different even for the same role depending on the context. Such approach to the goal and behavior description brings a possibility for agent to be more autonomous. Through utilization of this approach agent can change its role, set of the goals corresponding to its purpose depending on a condition of the environment. In other words, an agent can change its behavior based on a context.

Illustration of the concept of General Proactivity Framework is presented in Figure 10. We are using BDI (belief-desire-intention) model behind the framework in general and RgbDF in particular.

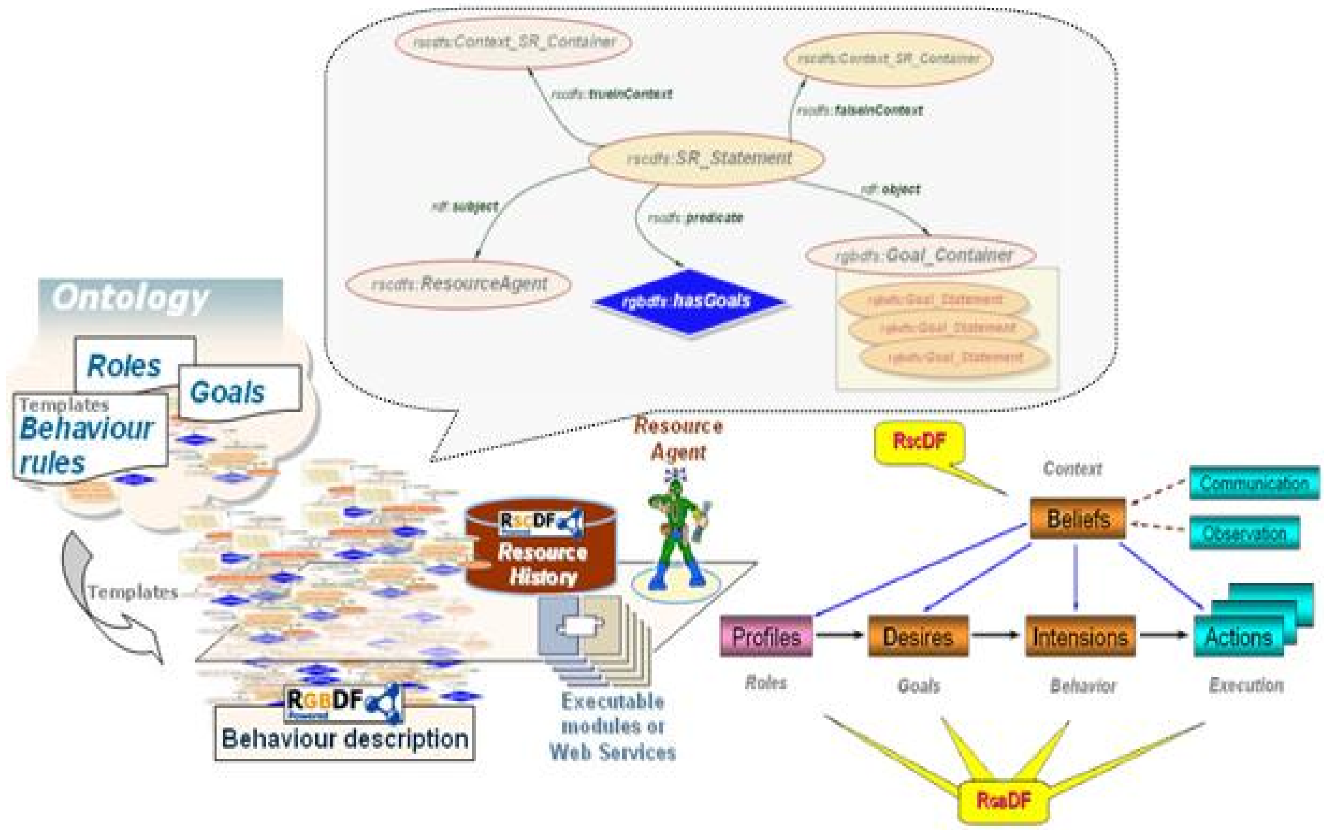


Figure 10 – The concept of General Proactivity Framework illustrated

The framework supposes annotating behavior of an agent (representative of some resource) with further enactment of the behavior. Thus, basically we faced two challenges: (a) design of a handy user interface for specifying resource behavior in form of rules and resource mental states, and (b) design of an engine to run these rules and perform corresponding actions. Such framework is anticipated to become a basis for modeling a variety of different processes: business processes, enterprise integration, distributed maintenance, distributed diagnostics and learning, supply chain management, etc.

The architecture of the proactivity layer of the SmartResource platform is presented in Figure 11. Its structure comprises four storages: (a) a history for storing facts about events occurred in an external environment of the resource agent; (b) a storage for reasoning (mental) states of the resource agent and rules that determine its behavior; (c) a storage where an ontology and all instances (metadata related to Devices, Services, Human Experts, Agents, etc.) are located; (d) a storage for programmable executable modules. In fact, the storage for the statements of facts about the external environment is presented by two storages: one for operational purposes and another for long-term storing. Operational storage contains relevant and up-to-date data critical for performance. For example, if a statement about assigning the resource agent a new role is asserted to the operational storage, then a statement about previous role of the agent must be removed to the long-term history or otherwise irrelevant alarm statements must be removed. Such filter prevents contradiction in the latest data.

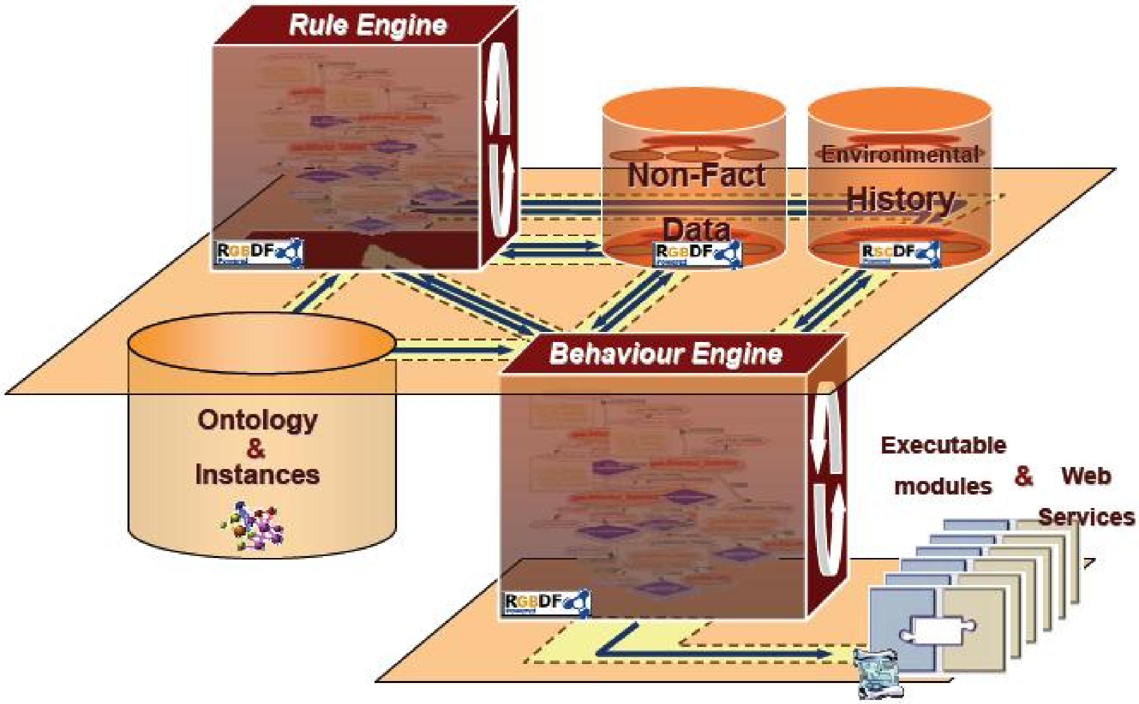


Figure 11 – Architecture of the Proactivity Layer of the SmartResource platform

The architecture includes also two engines: one for rules and another for a behavior. The engines iteratively check the rules, execute them and launch necessary actors (modules). The main role of the rule engine is to generate (to change) a context for the resource behavior.

From the user's perspective, the information that will be needed from him/her is (a) a starting role or goal for the resource agent and (b) input data/facts. For this purpose, the user interface must provide all available information from the ontology and data, stored on the platform: a list of instances, a list of properties, etc. If semantic profiles of accessible executable modules and web services are available, then this makes a good basis for automated generating behavioral rules by the platform. Otherwise, a semantic profile has to be specified for all executable modules available on the platform and for web services that will be used. If there is no any executable module or a web service, which can achieve the goal, then the goal can be divided into a set of sub-goals based on corresponding information in the ontology or on an iterative process of generating sub-goals automatically (based on required inputs for modules, which can reach the goal but inputs are not provided).

Ontology-driven approach in modeling agent behavior as context-sensitive dynamic change of standardized and reusable roles, goals and actions, anticipated to become a reasonable solution providing some benefits comparably to conventional model-driven approaches. Resource Goal/Behaviour Description Framework was a natural stage of development of a semantic modeling basis for the overall SmartResource platform.

In some cases, it is better to utilize other (well used for specific domain) behavior execution engines. For example, in case of process performed by web services, it makes a sense to use Business Process Execution Language (BPEL) Workflow Engine. With this purpose we have a need to enhance the Platform with Transformation module that transforms RgbDF behavior description to BPEL description of a process. Advantage of RgbDF process representation is that web services can be described through semantic profiles instead of exact web service description. Thus, we make a step from individual WS binding to Semantic Scenarios Specification. It gives a possibility to select suitable web services from the available set, and just after that to make a transformation to BPEL scenario. Such approach brings possibility to share and utilize knowledge about process without dependence on available services (Figure 12).

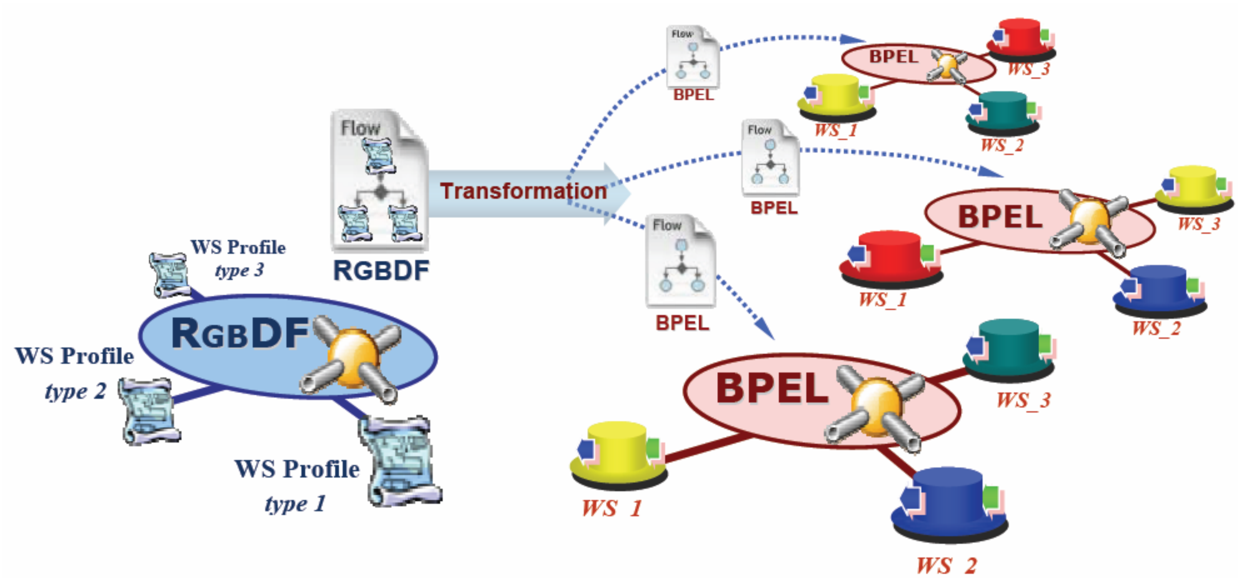


Figure 12 – Resource-independent process knowledge sharing

Domain ontology provides common shared understanding of the domain representation. Each web service is semantically annotated according to this ontology by means of a WS Profile. Business Processes are modeled in a implementation-independent way (e.g. without hard binding to concrete service implementations) and can be stored in RgbDF or in another suitable process modeling language which allows for decoupling of business process logic from concrete activity implementation. Business process in this case represents rather logic of semantic data flow between semantically described service profiles. Real world web services can be considered as instances of the corresponding web service profiles (e.g. objects of a class in OOP). The flow enactment can be done dynamically by selecting Semantic Scenario Specification and automatically transforming it to ready-to-execute BPEL file. The transformation procedure lies in the selection of instances of appropriate web service profiles involved in a particular scenario (in MDA terms: Platform Independent to Platform Specific Model transformation).

There are quite many other activities in Agent Behavior area. One of those is an initiative of France Telecom Research & Development (FTR&D). They provided JADE Semantics Add-on as a framework based upon JADE (Java Agent Development Framework) to interpret the meaning of the exchanged speech acts, according to their formal semantics as specified by FIPA-ACL; to make agents more flexible, in order to better interact in open environments; to simplify the coding of JADE agents.

5. General Networking Framework

The next important challenge of GUN in general and SmartResource in particular is to make every domain resource collaborative, which means on the one hand coordination of autonomous and proactive parts of this resource (which are also smart resources themselves) and on the other hand coordinate own behavior with other resources within an organization towards achieving consensus between personal and organizational goals. General Networking Framework (GNF) should provide tools to design, share and reuse universal semantically-configurable scenarios for required coordination (Figure 13).

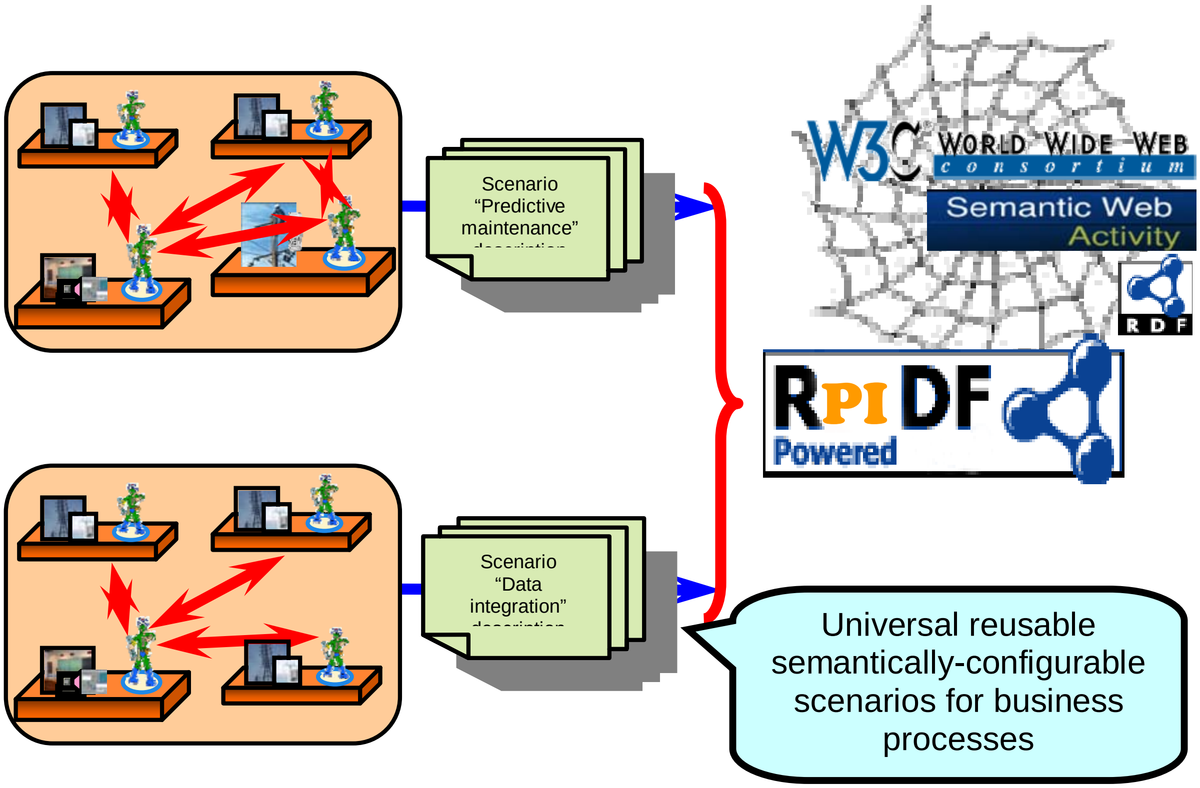


Figure 13 - The challenge of universal self-configurable scenarios for coordination

GNF is also a technology and a platform for integration individual behaviors of proactive smart resources into a business process with opportunity to manage the reliability of components by certification, personal trust evaluations and exchange.

The General Networking Framework considers an opportunity of ontological modeling of business processes as integration of component behavioral models of various business actors (agents representing smart resources in the web) in such a way that this integration will constitute the behavioral model of an agent responsible for the *alliance* of the components. This means that such corporate agent will monitor behaviors of the proactive components against the constraints provided by the integration scenario. Such model is naturally recursive and this means that the corporate agent can be a component in a more complex business process and will be monitored itself by an agent from the more higher level of hierarchy. Hierarchy of agents can be considered as possible mapping from the part-of ontological hierarchy of the domain resources (see Figure 14).

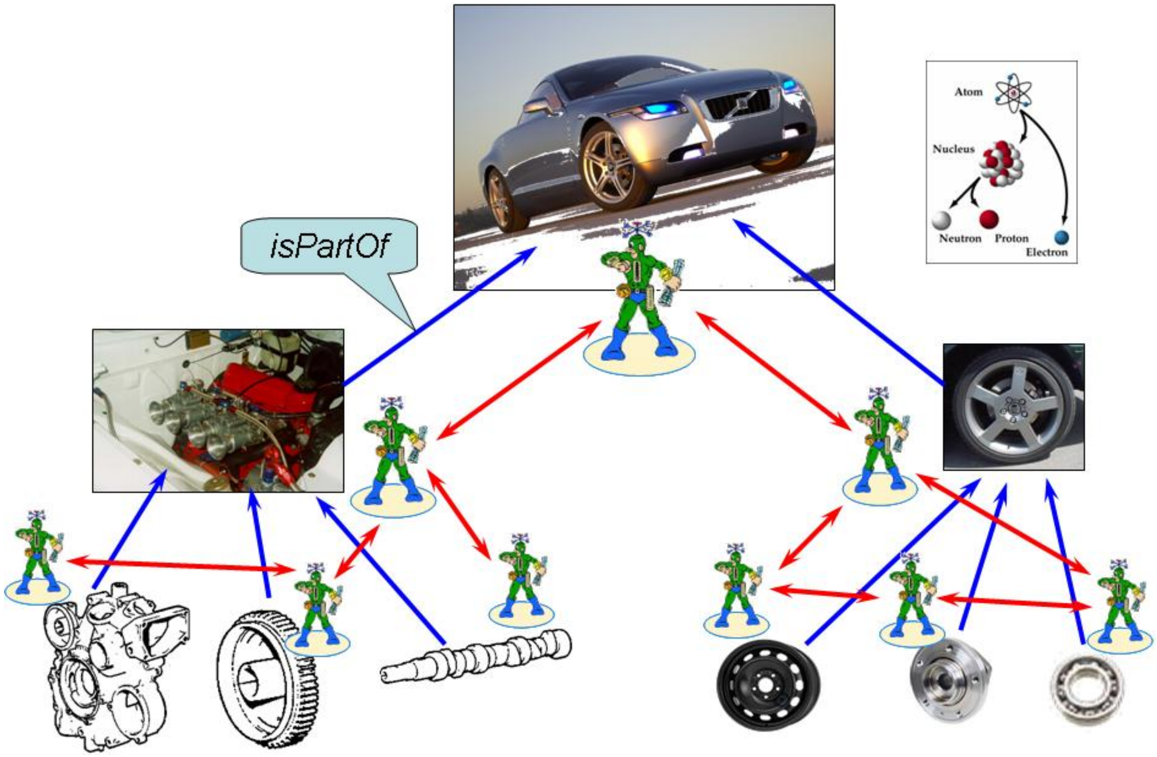


Figure 14 - Part-of hierarchy of resources results in corresponding hierarchy of agents

Another important concern is: What is a process in GUN environment? Consider the following two axioms of GUN:

Axiom 1: Each resource in a dynamic Industrial World is a process and each process in this world is a resource.
Axiom 2: Hierarchy of subordination among resource agents in GUN corresponds to the *part-of* hierarchy of the Industrial World resources.

As all GUN resources, a process has own properties that describe process's state, history, sub processes and belongingness to upper-process (super-process). Thus, following the principles of GUN resource, each process should be enhanced with an Agent that serves this process as well as to any other resource. GUN's Top Agent is the one, which resource, to be taken care of, is the Industrial World as whole. Such agent will be on the top oh the hierarchy of resource agents.

Each industrial resource can theoretically be involved to several processes, appropriate commitments and activities, which can be either supplementary or contradictory. This means that the resource is part of several more complex resources and its role within each of the resource might be different. Modeling such resources with GUN can be provided by appropriate resource agent, which can make clones of it and distribute all necessary roles among them (see Figure 15).

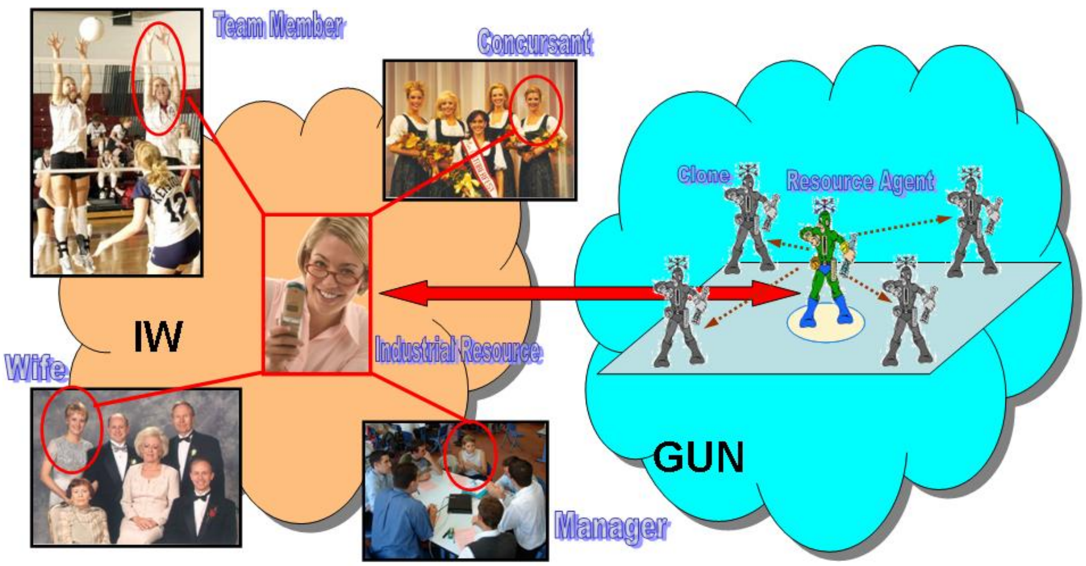


Figure 15 - Multiple roles of a resource in the Industrial World and appropriate agent-clones in GUN

Each industrial resource, which joins some commitment, will behave according to restrictions the rules of that commitment require. The more commitments individual resource takes, the more restriction will be put on its behavior (see Figure 16).

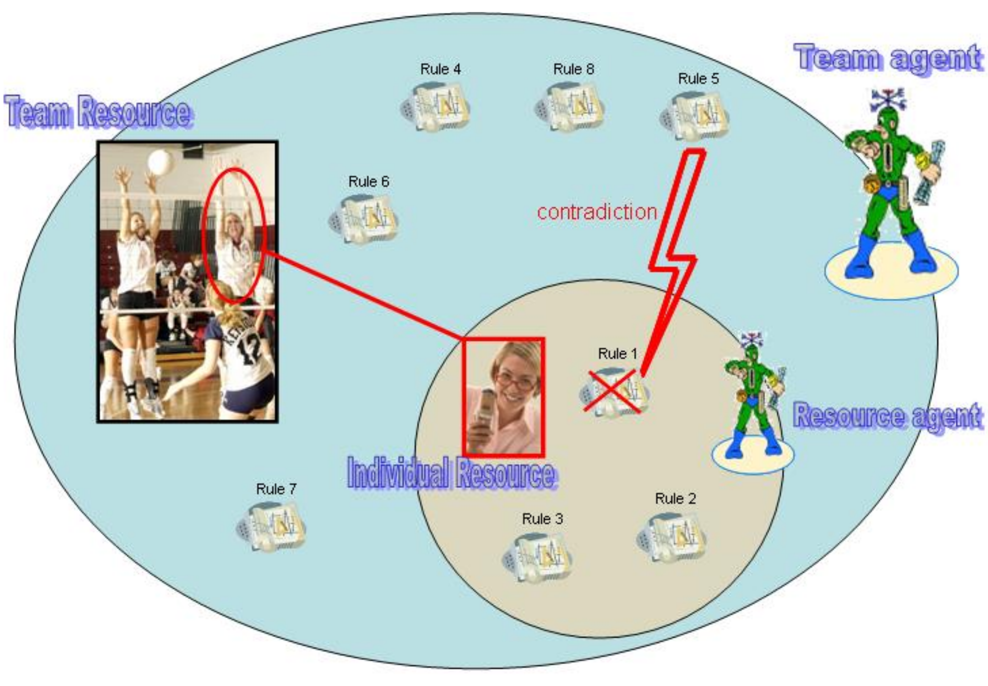


Figure 16 - Individual vs. team resource freedom

The main feature of the General Networking Framework is smart way of managing commitments (processes and contracts) of any proactive world resource (SmartResource) to enable cooperative behavior of it towards reaching also group goals together with the individual ones. Taking into account that world of industrial products and processes has multilevel hierarchy (based on *part_of* relation), we can say that it results to a hierarchical structure of GUN agents, which are meant to monitor appropriate world components in a cooperative manner (see Figure 17).

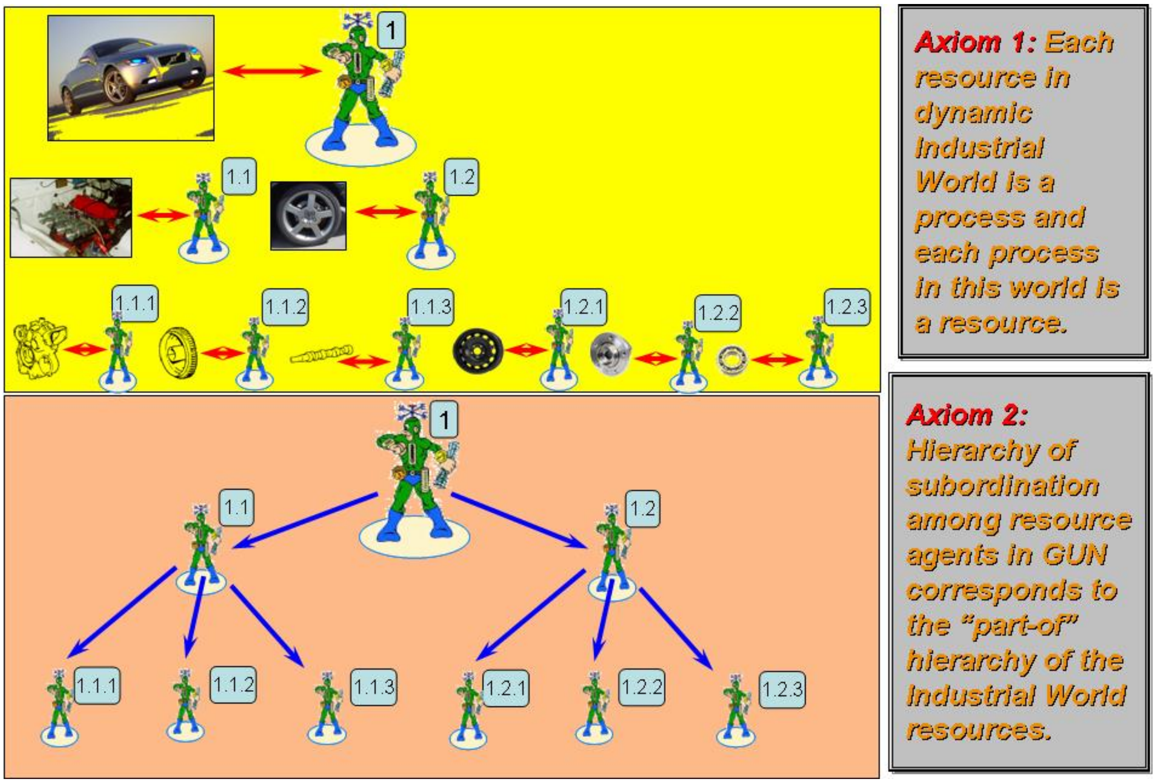


Figure 17 - Agent subordination according to GUN axioms

Summarizing we can say that GUN vision assumes proactivity of all heterogeneous resources (humans, devices, services) in the World of Things and intends to provide reusable behaviors and reusable coordination patterns to all the resources making them components in a self-organized process of automatic creation of complex dynamic reconfigurable systems for different industrial purposes. GUN vision allows considering everything as a smart agent-driven resource, which are not only physical objects, but also processes, mathematical models, ontologies and even messages in communication. The last one allows dynamic (smart) routing, where a smart message itself (not the nodes) decides where to go further within a network, is able to collect own history and communicate with others.

6. Other Challenges of the Global Understanding Environment

GUN vision being very general has quite many opportunities to be checked and open issues to be studied. Below some of the GUN challenges are listed, which has not been studied deeply yet or only preliminary studies are available.

6.1. Semantic Annotation, Exchange, Integration and Mobility of Diagnostic Models and Decision Engines

As it was mentioned above the GUN environment is supposed to have decision making resources, which are based on certain machine learning algorithms. Getting data from some external industrial resources (devices, machines, etc.), such algorithms are able to build models for diagnostics and performance prediction of these devices. Natural heterogeneity and distribution of these algorithms and models result to another important challenge of GUN environment, which is to provide an opportunity for automated algorithms (learning and decision engines) and models discovery, sharing, reuse, interoperability, invocation, integration and composition. For that some ontology and markup for learning algorithms should be developed as well as the ontology and markup for learned models.

Example is shown in following pictures. Let some Bayesian Network (Figure 18) has been learned based on training data from some industrial device. The appropriate GUN resource (e.g. Web-Service) has produced this model according to Bayesian learning algorithm and now keeps it in some internal format (e.g. as it is shown in the picture). It is evident that other external services (even the ones, which also provide Bayesian learning and reasoning) will not necessary "understand" that model because of possibly different format of representation. Imagine that two heterogeneous Bayesian reasoners have studied the same object based on different subsets of data and got different Bayesian models as a result. Is there any way to integrate these two models without re-learning so that each of the reasoners can use the more advanced integrated model? This can be done if the representation of both models is done according to the same standard. Just for that it is reasonable to develop semantic markup (e.g. as shown in the picture) for various models representation as well as the ontology of models itself. In this case an integration of the models from similar class can in principle be done automatically without involvement of the original training data.

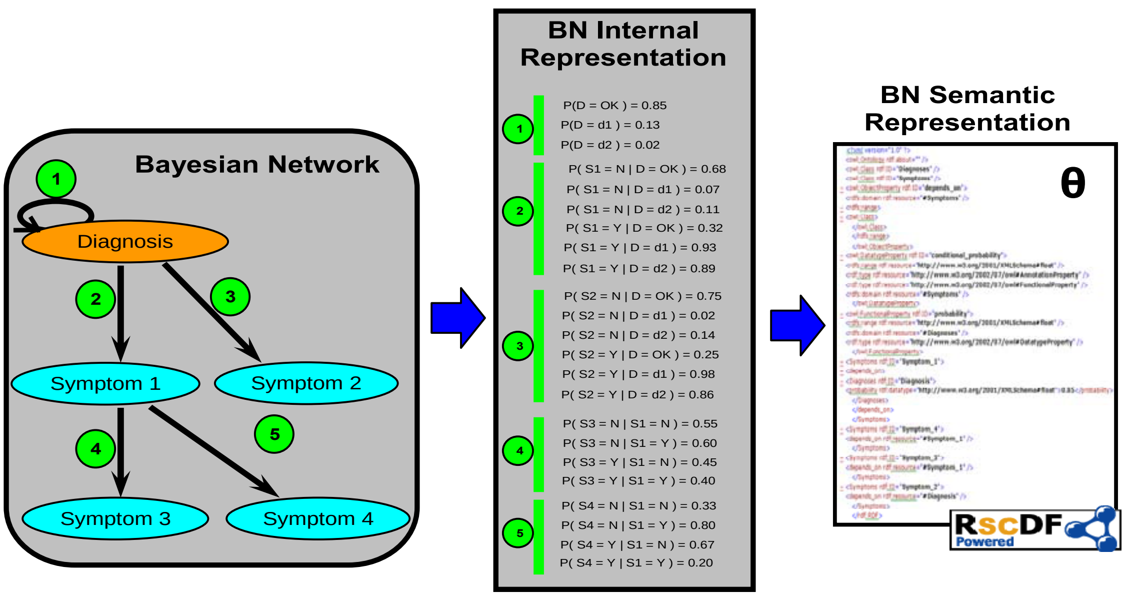


Figure 18 - Sample of a Bayesian Network model (graphical, formal symbolic and semantic representations)

The problem of integration becomes much more complicated if involves the models from different classes. For example, the same part of object data was observed by Bayesian learner on one hand, and on the other hand some other part of the same object data was observed by Neural Network learner (see Figure 19). Even if both models were semantically mark-upped and integrated, to be able to run such a hybrid neither Bayesian reasoning engine nor Neural Network reasoning engine can be applied. This means that to enable to run integrated models a universal reasoning engine should be developed, which will be able to integrate semantically annotated heterogeneous engines. The sample of Bayesian Network reasoning engine internal and semantic representation is given in Figure 20.

Thus the challenge discussed above includes ontology and markup for learned models description, ontology and markup for the decision (reasoning) engines description and appropriate tools for models and engines automatic discovery, exchange, integration or composition.

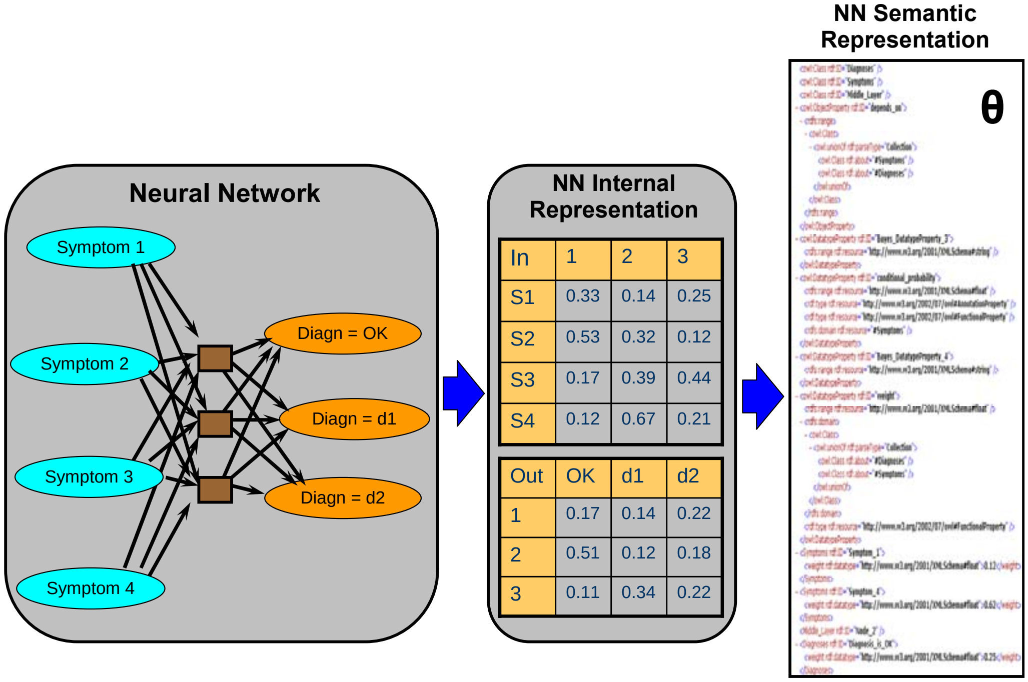


Figure 19 - Sample of a Neural Network model (graphical, formal symbolic and semantic representations)

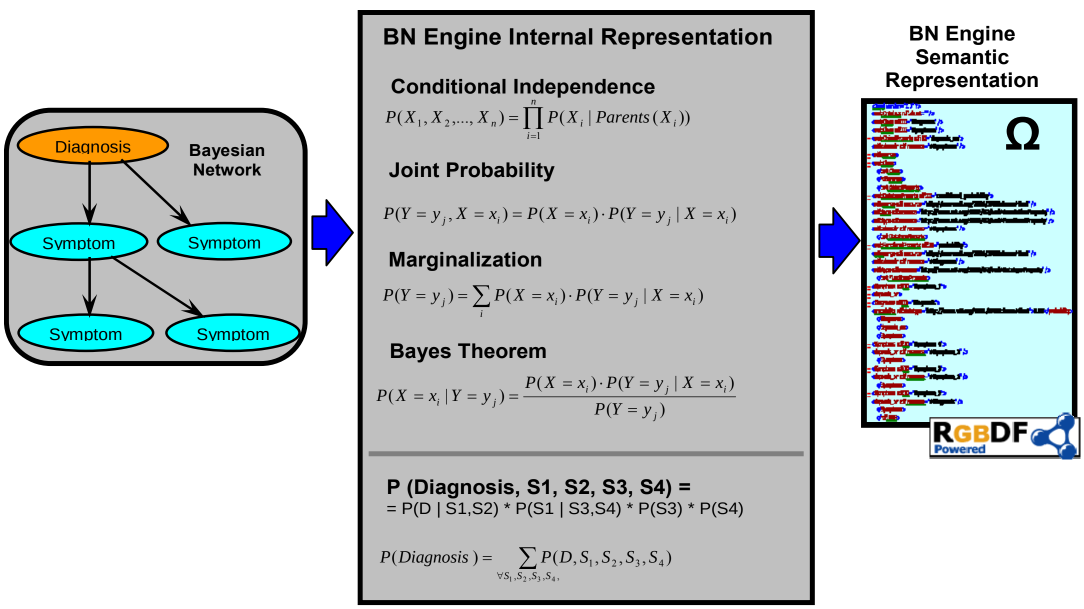


Figure 20 - Sample of a Bayesian Network reasoning engine internal and semantic representation

Such implementation may allow also a mobility of the decision-making entities within GUN network. Imagine a mobile agent, which functionality is to move from platform, learn models from different sources and implement learned models later also in different places (see example in Figure 21).

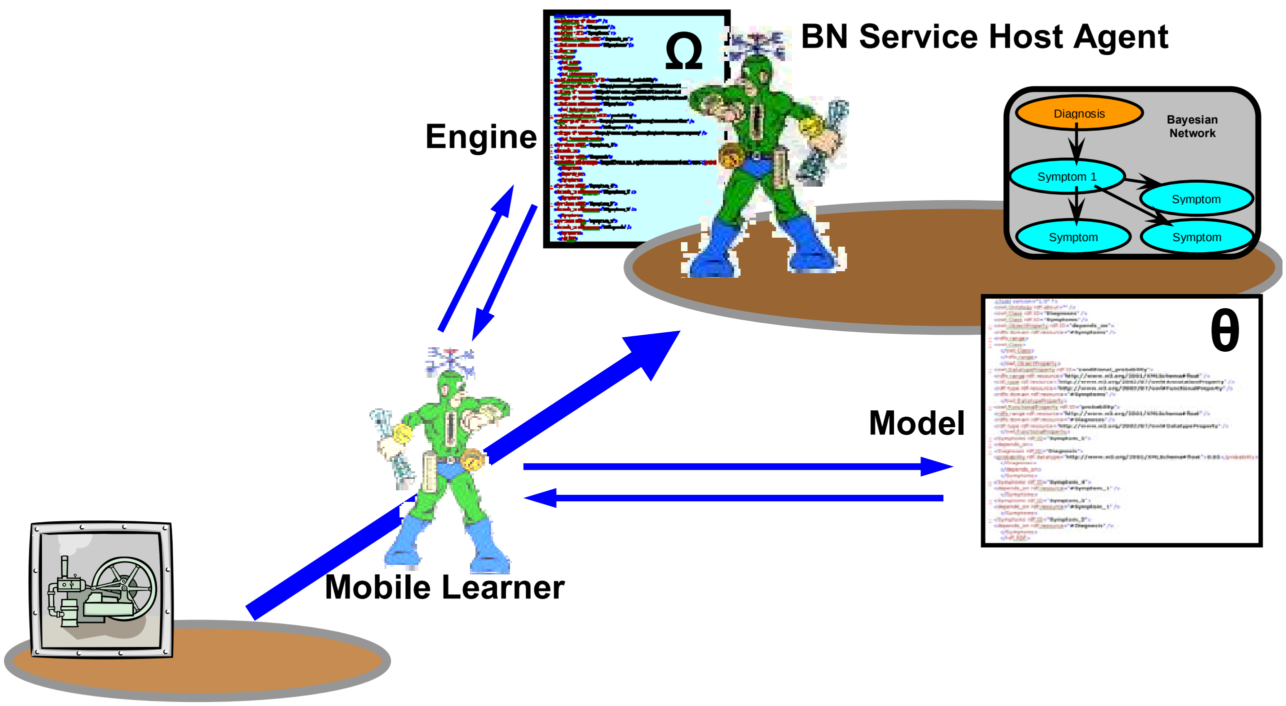


Figure 21 - "Mobile Learner" in GUN environment

Such agent:

- Is a shell (engine);
- Can learn a learning algorithm (metalearning);
- Can learn a model based on data and learning algorithm;
- Can "swallow" and run existing model;
- Can learn contexts;
- Can reason based on model, data and context;
- Can move to "universities" (special platforms with appropriate Web-Services) for learning;
- Self-interested;
- Can take care of a GUN resource;
- Its learning ability is ontology-based.

There is also sense to talk about "mobile models", which is similar but not the same concept as the mobile learner. A mobile model is useful for various personalized web services for mobile users. User data is being collected in the mobile client terminal and by portions goes to server, server learns a model based on this data, model "goes" back to the mobile client and run on it. The sample scenario can be as follows: (1) the log of online data from some GUN resource is collected locally; (2) collected data after semantic enrichment is sent to special machine learning Web-service (e.g. for Bayesian learning as shown in Figure 22); (3) based on this data the service creates a model and transfers it to semantic format (Figure 23); (4) nominated by the platform agent with learned model and appropriate engine moves to the original client platform (Figure 24); (5) the "guest" agent will perform further diagnostics of the original resource locally.

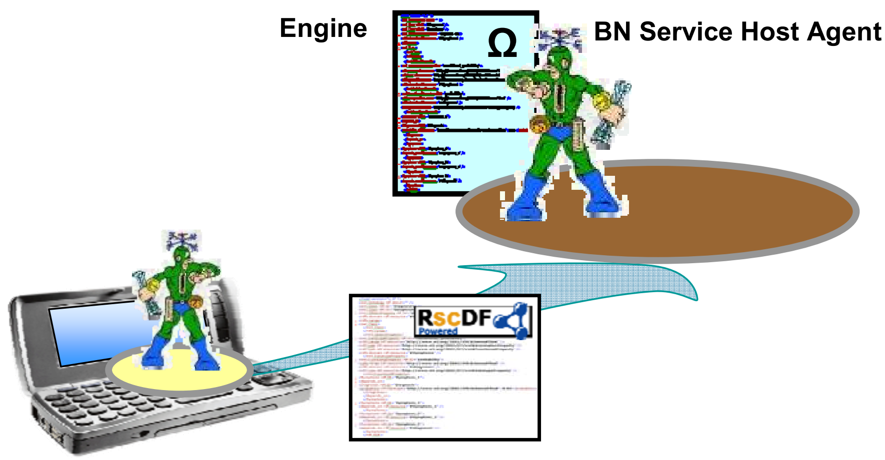


Figure 22 - Sample scenario for "Mobile Model" concept: annotated resource data is sent to a service

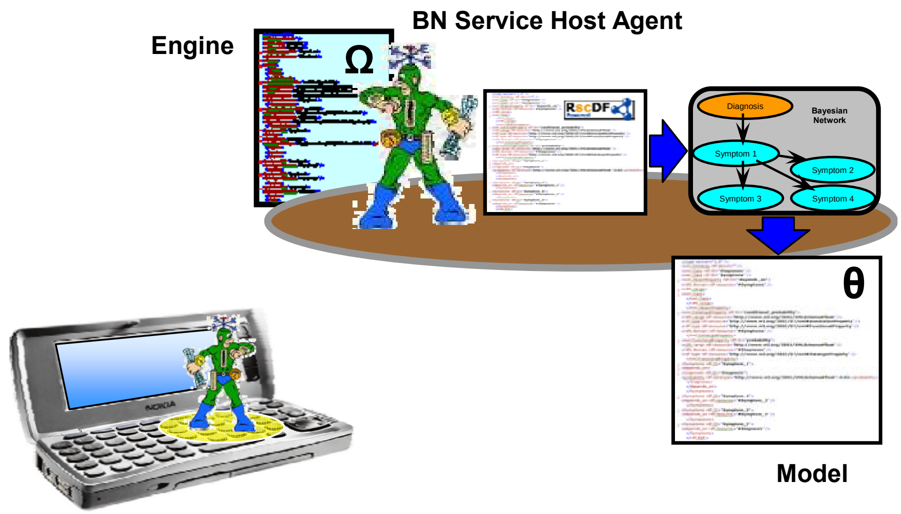


Figure 23 - Sample scenario for "Mobile Model" concept: service learns a model from data

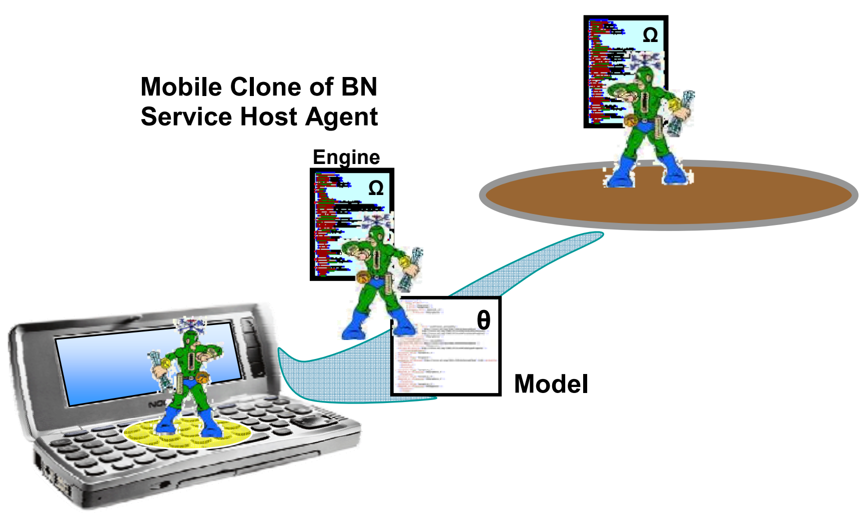


Figure 24 - Sample scenario for "Mobile Model" concept: agent with model and engine moves to the original client platform

6.2. Learning Distributed Trust

Another challenge is to manage "trust" in GUN environment. This is especially important due to extensive use of numerous decision making services (for condition monitoring, diagnostics, prediction, maintenance planning, etc.). In cases when a resource is making choice to which decision making service it should send its query, the knowledge about services performance in past would be quite important. Trust of some service requestor to some service provider can be considered as personalized prediction of service performance (quality, precision, etc.) partly based on the success history of previous interactions between the requestor and the provider and also partly based on known or communicated trust evaluations from other requestors or trust certification authorities.

The challenge of learning distributed trust contains:

- Automated certification of GUN Resources;
- Trust calculation;
- Trust monitoring, diagnostics and maintenance;
- Trust models learning;
- Trust prediction;
- Trust sharing and exchange;
- Smart P2P environments based on trust;
- Trust-related security and privacy issues.

Consider sample scenario from Figure 25. Assume some industrial GUN resource (device) has collected on a local GUN platform labeled data (e.g. personal history faulty states descriptions labeled by experts with appropriate diagnoses). The following procedure is the one widely used in machine learning field known as learning ensembles of classifiers. The local platform agent divides the data into two subsets: training sample set and test sample set. The training set is given to several external services so that they are able to adjust their models or learn the new diagnostic model specifically for that device from the scratch. External services can in principle support different learning algorithms (e.g. Neural, Fuzzy, Bayesian, Genetic, etc.). After all contacted services report that they have learned, the device provides them another part of labeled data, i.e. training set with hidden actual diagnoses. Services are requested to provide the diagnoses for given cases. Device agent compares received outcomes from services with known (actual) diagnoses and calculates for each service the performance value (e.g. percentage of correctly classified cases from the test set) and considers these values as personalized ranks of services, which determine the trust of the "device" towards appropriate service providers. The ranks can be used in future to diagnose new states of the device either by selecting the best ranked service from available ones or my integrating outcomes from several services as weighted (based on ranks) voting among them. This approach should increase diagnostic performance in comparison to randomly selected service or simple averaging of outcomes from several services.

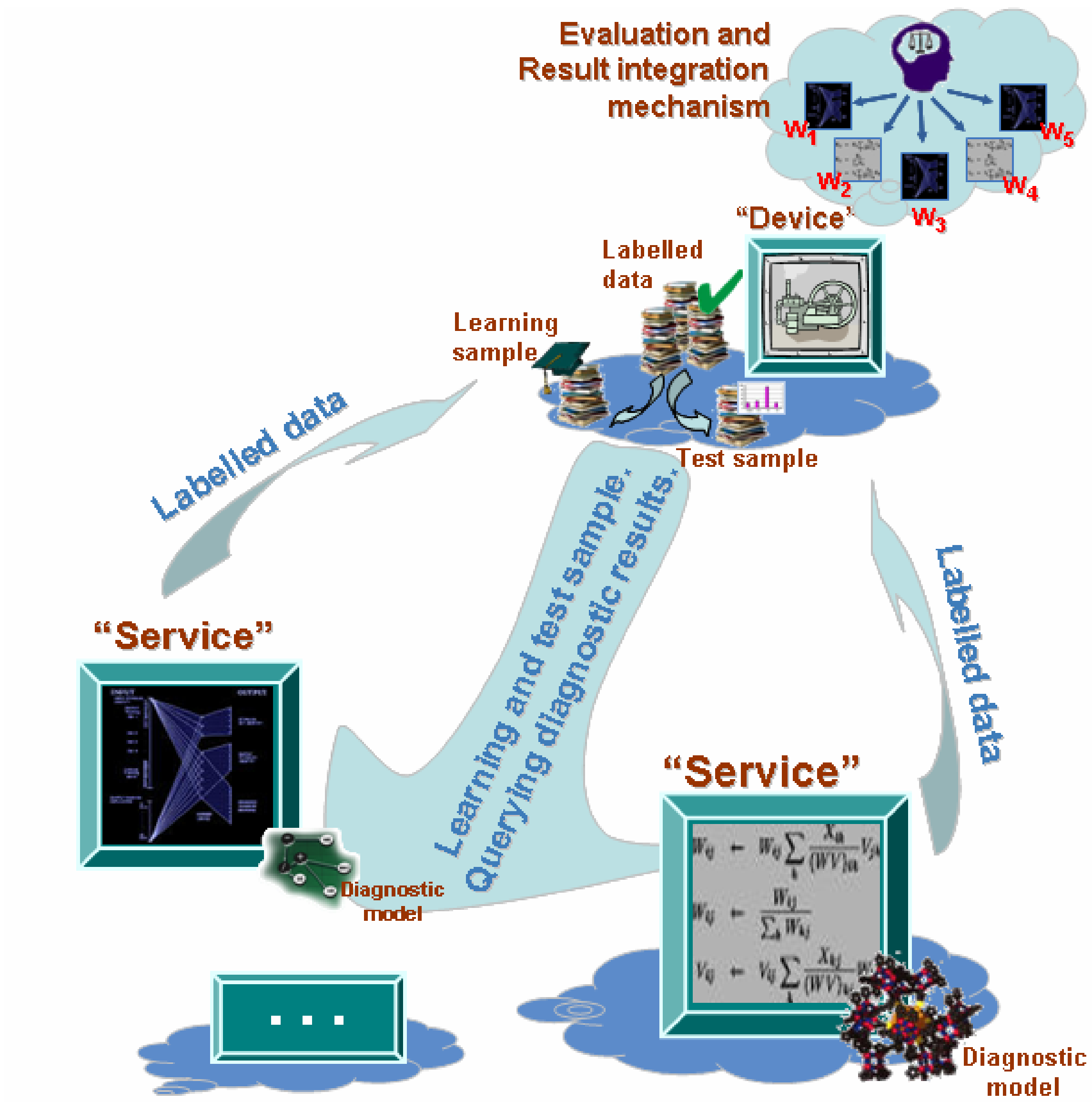


Figure 25 - Sample scenario for "Learning Distributed Trust" concept: service learns trust values by providing training and test samples of data for various services to use these values later for smart service integration

Consider another scenario in Figure 26. Here in addition to "personal" trust evaluation, the client (e.g. industrial device) can use also professional evaluation of services performance from special certification service. It is reasonable that such certification service is provided by the device manufacturer to check all external services, which pretend to provide reliable diagnostics. The advantage of the evaluation provided by a certification service is availability of quite complete set of training and testing data collected from a number of the same kind devices. However it is also true that the advantage of personal evaluation (previous scenario) is based on personalized model creation (data might be incomplete but belongs to own history).

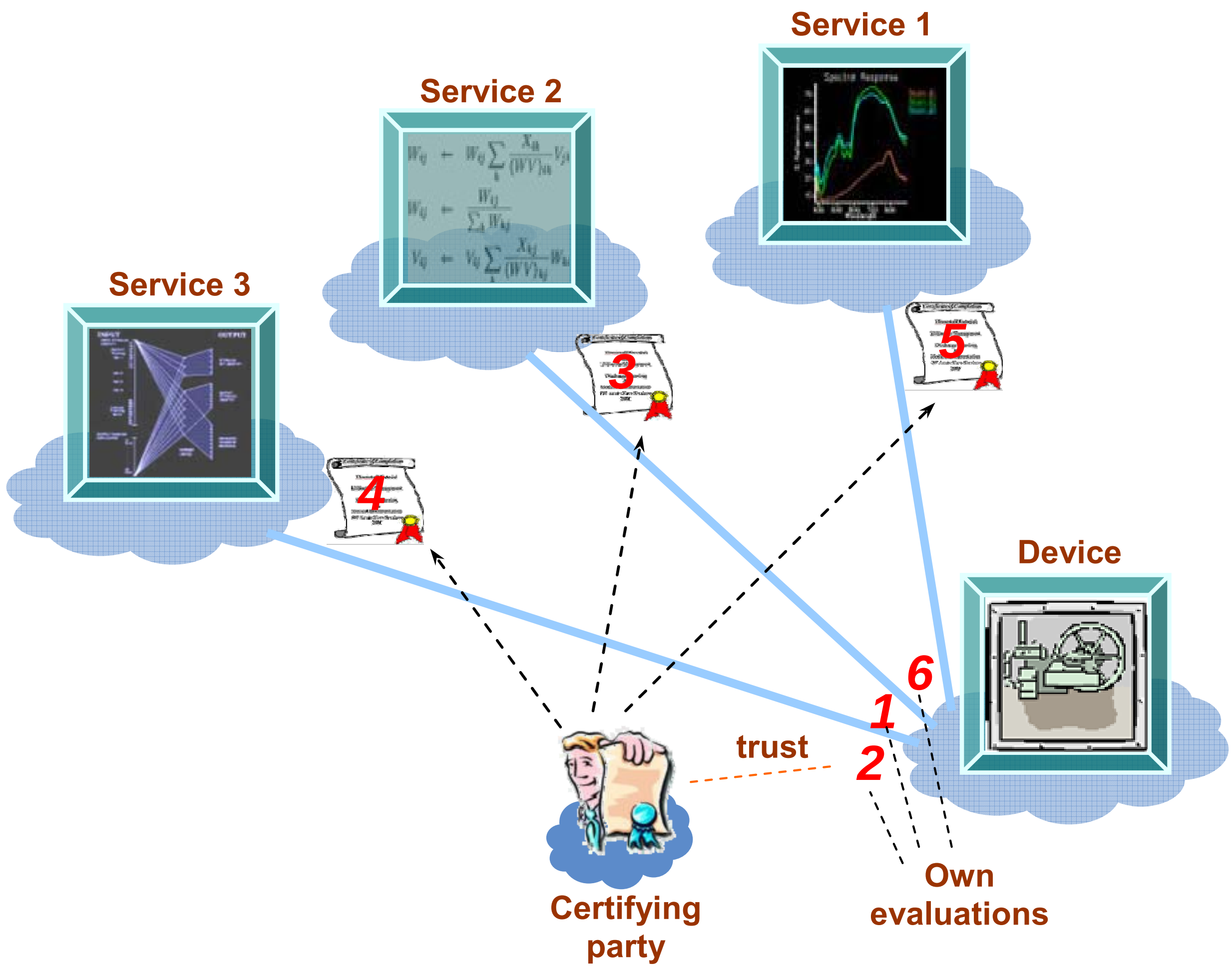


Figure 26 - Sample scenario for "Certification Service" concept: certification service learns trust values by providing complete training and test samples of data. Clients (devices) can access certification service and request trust certificate concerning service they are going to use. In combination with previous scenario, device can combine personal trust values and certified values to compute rank of a service.

Consider one more scenario in Figure 27. Here the case is shown when some client (device) has not done personal evaluations of some services but trying to get such values from the "colleagues" in the network (other devices of the same type, which already evaluated the services in question). The trust towards the colleagues itself should be taken into account and combined to the values of trust towards services they share.

Actually the complete trust evaluation of a client towards some service is some kind of smart combination of (a) personal evaluation, (b) evaluation obtained from a certification service, (c) evaluation communicated from the colleagues (clients of the same type).

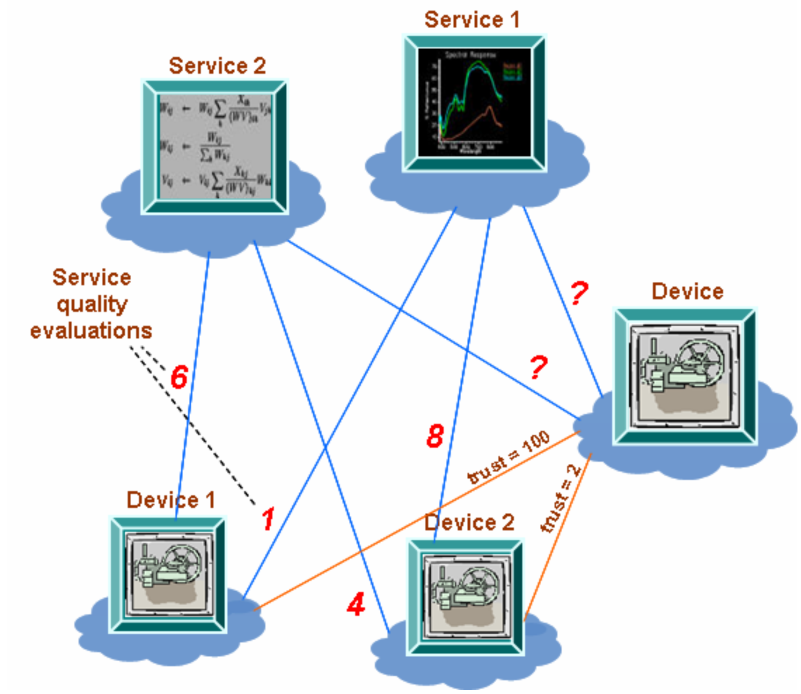


Figure 27 - Sample scenario for "Trust exchange" concept: the device imports trust values concerning services 1 and 2 from the "colleagues" (Devices 1 and 2). The device should also take into account the trust values towards colleagues if known.

6.3. Human as a Web-Service

Another interesting challenge of GUN is that the role of Web-service can be played by a human. Because different categories of industrial resources (e.g. devices and maintenance experts) can be adapted to GUN environment with adapters and provided by agents, it may happen that the device can be a service requestor (e.g. require diagnostic based on measured state of parameters) and a human can be a service provider (e.g. the one who can online provide a diagnosis based on state description). The challenge here is how to take into account human specifics to mange such architectures (as e.g. in Figure 28) in a reliable way and also how to enable automated discovery, sharing, reuse, interoperability, invocation, integration and composition of online human resources in the Web.

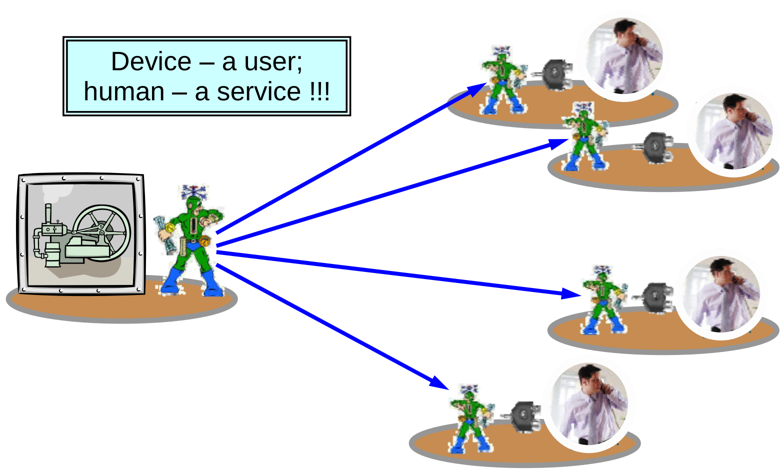


Figure 28 - "Human as a Web-Service" concept

7. RDF Evolution towards Context Description Framework

Recent expectations regarding the new generation of Web strongly depend on the success of Semantic Web technology. Resource Description Framework (RDF) is a basis for an explicit and machine-readable representation of semantics of various Web resources and an enabling framework for interoperability of future Semantic Web-based applications. RDF is not suitable for describing highly dynamic and context-sensitive resources (e.g. industrial devices, processes, etc.). Also RDF lacks tools to describe autonomous and proactive resources, processes and scenarios. That is why we considered important to extend RDF towards making it enable to describe smart resources in general. Such extension should base on RDF syntax and only extend semantics appropriately. Our research on GUN in the SmartResource project has pointed out the need of updating RDF as the basic Semantic Web framework – in three dimensions: regarding context-sensitivity and dynamics, proactivity, and coordination (see Figure 29).

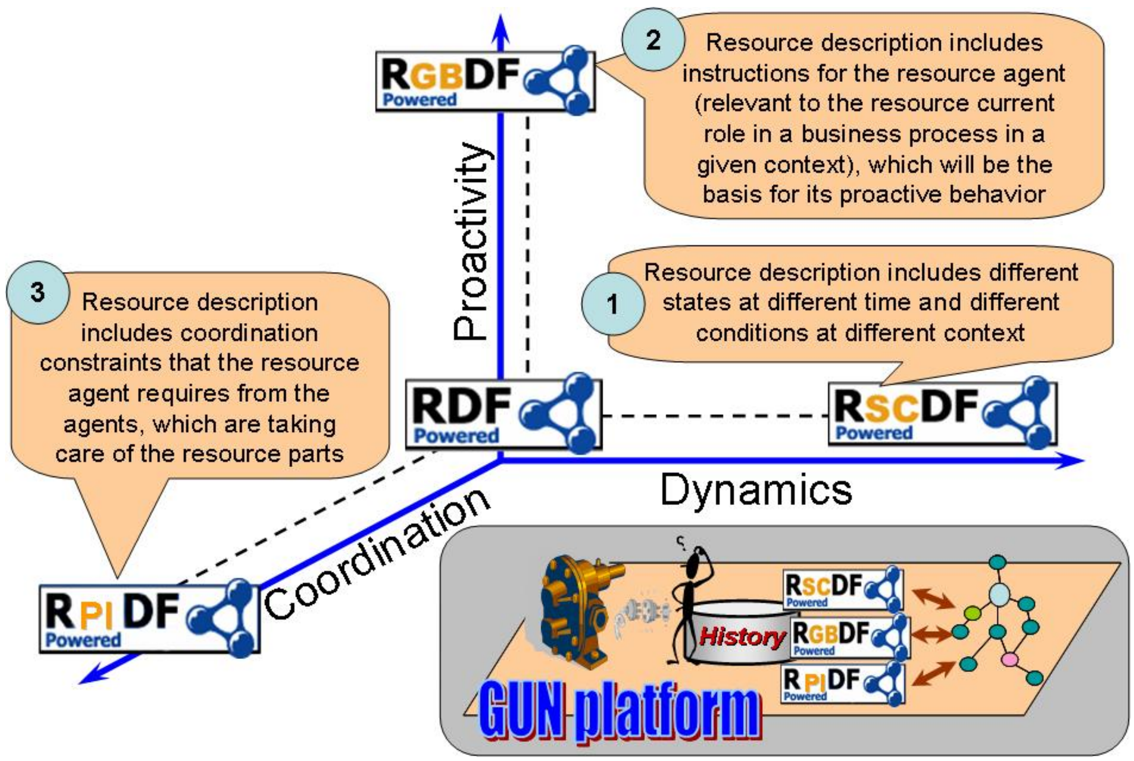


Figure 29 - Three dimensions of developing RDF towards the Web-of Things domain

The integrated view to all three extensions (RscDF, RgbDF and RpiDF) is based on the same framework, called Context Description Framework (CDF) as a logical extension of the existing RDF. We add a "TrueInContext" component to the basic RDF triple ("subject-predicate-object") and consider contextual value as a container of RDF statements. We also add a probabilistic component to the model, which allows for multilevel contextual dependence descriptions as well as presumes possibility for Bayesian reasoning with RDF model.

In our vision all properties (predicates in RDF statements) have certain sense in a certain context which should be specified by the context tolerance range. Thus we have a need to define a contextual range for a property in the ontology. Each RDF statement may be true or false concerning the different conditions of an environment. In this case we consider the context of a statement as a set of other statements that describe a certain condition (state) of an environment. Such descriptions among properties of an environment may contain also the source of the statement descriptions, and thus provide opportunity to manage trust in distributed systems. Each contextual statement itself may also have its own context (i.e. a nested context). A nested context provides new possibilities for vertical in-depth reasoning based on context-sensitive descriptions. We found out that using a triplet-based model for a statement-in-context description is not suitable, and therefore use quadruples for modeling, where the fourth component is a container of contextual statements.

A Context Description Framework is a logical extension of the RDF and is meant to model the context dependence of the world properties. It allows us make two significant steps in the resource description approach. We logically go from a duplet (domain-range) vision of a property description in ontology to a triplet description (domain-range-context), and from a triple representation of a statement to quadruple representation (statement in a context of other statements).

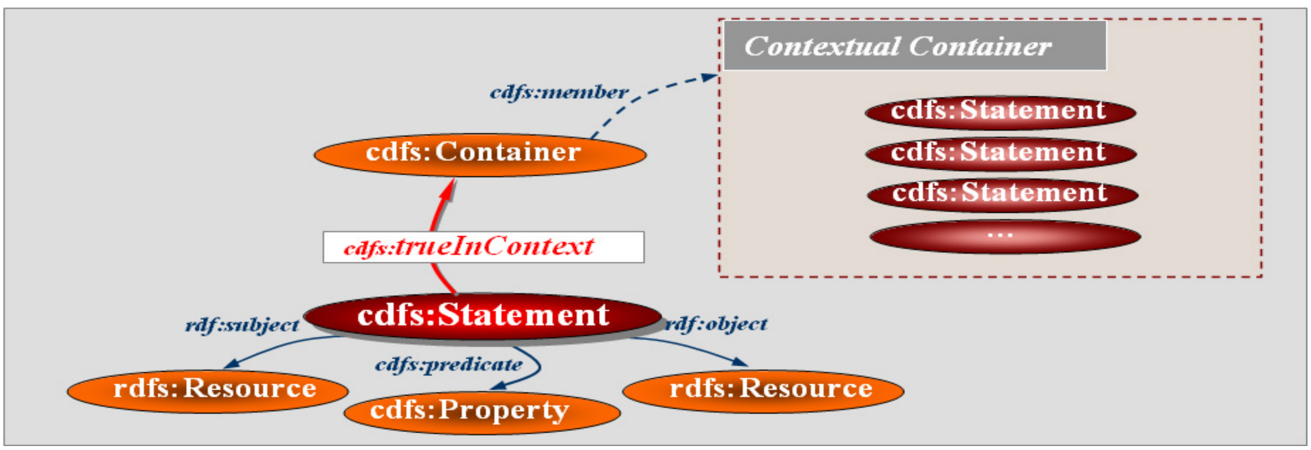


Figure 30 - A quadruple vision of the statement

Concerning the second significant step (the extension to a quadruple statement representation), we define a CDF quadruple (see Figure 30). A CDF quadruple contains four components: a subject, which is an RDF URI reference or a blank node; a predicate, which is an RDF URI reference; an object, which is an RDF URI reference, a literal or a blank node; and a contextual container (context), which is an RDF URI reference or a blank node. A CDF quadruple is conventionally written in the following order: subject, predicate, object, contextual container. A predicate is also known as the property of a quadruple. With a purpose to define a CDF quadruple we have inherited the rdf:Statement class and have added the additional cdfs:trueInContext property. A CDF statement is a statement made by a token of a CDF quadruple. A subject of a CDF statement is an instance of rdfs:Resource identified by the subject of the quadruple. A predicate of a CDF statement is an instance of cdfs:Property identified by the predicate of the quadruple. An object of a CDF statement is an instance of rdfs:Resource identified by the object of the quadruple. A context of a CDF statement is an instance of cdfs:Container identified by the contextual container of the quadruple. The cdfs:trueInContext property has the cdfs:Statement and cdfs:Container classes as the domain and range accordingly, where the cdfs:Container class is inherited class from the rdfs:Container and restricted with a content. The instances of the cdfs:Container class may contain only the instances of the cdfs:Statement class, which play a role of a statement context. Figure 30 shows a quadruple approach to statement representation. At the time we create the cdfs:trueInContext property, we also add a similar cdfs:falseInContext property to describe the context within which the subject statement is false. Now we can describe any statement with a binding to a context. Such a context-dependent representation of a statement entails a specification of the contextual container content range accordingly to a quadruple predicate. Thus we come to the necessity of making one more logical step in the resource description approach and go to a triple vision of a property.

Following the first step we extend an existing rdf:Property, which is described by rdf:domain and rdf:range, with a crdfs:context description (exactly with a “context tolerance range” definition). As the RDF Concepts and Abstract Syntax specification describes the concept of an RDF property, we describe the concept of a CDF property as a context-dependent relation between the subject resources and the object resources. A CDF triple property representation contains three components: a domain, which refers to a domain class; a range, which refers to a range class; and a context, which refers to a set of the contextual properties (context range). Figure 31 shows a new triple vision of a property. As a rdf:domain property defines a restricted area (rdfs:Class) of the subject property domain and a rdf:range property sets a subject property range (rdfs:Class), cdfs:context property defines a vector of the properties (cdfs:ContextContainer) that play role of a subject property context.

Class cdfs:ContextContainer is a subclass of the rdfs:Container in a general case. It contains a set of the cdfs:Property instances. They restrict the number of properties that can be used as the objects of a cdfs:predicate property in a contextual statement description. In other words, this container specifies a range (set of the object properties) for the cdfs:predicate properties of the statements in contextual container of the subject Statement (Figure 32).

Due to the new vision of a resource description, we redefine the concept of a subproperty. The cdfs:subPropertyOf property may be used to state that one property is a subproperty of another one. If a property P is a subproperty of property P', then all triplets of resources (subject resource, object resource, and trueInContext container) that are related by P are also related by P'. The term super-property is often used as the inverse of subproperty. If a property P' is a super-property of a property P, then all triplets of resources that are related by P will be also related by P'.

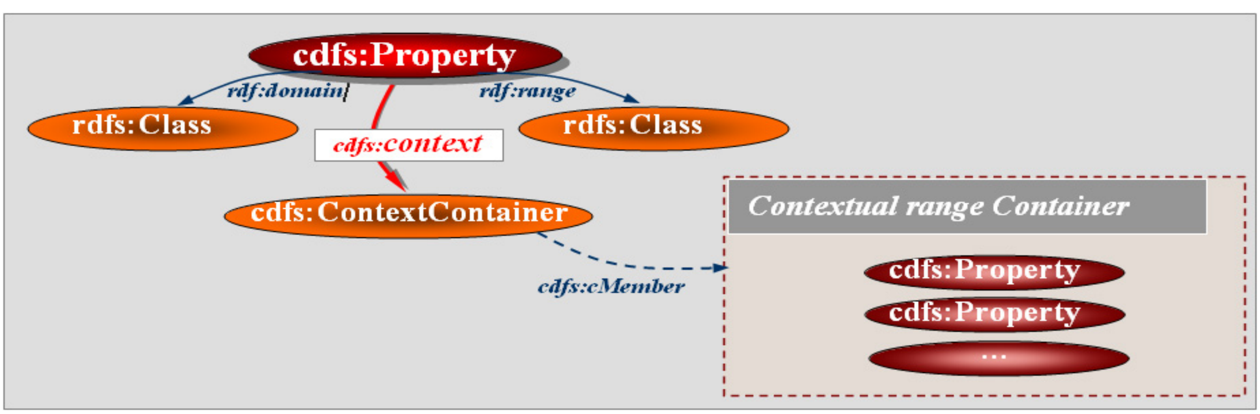


Figure 31 - A triplet vision of the property

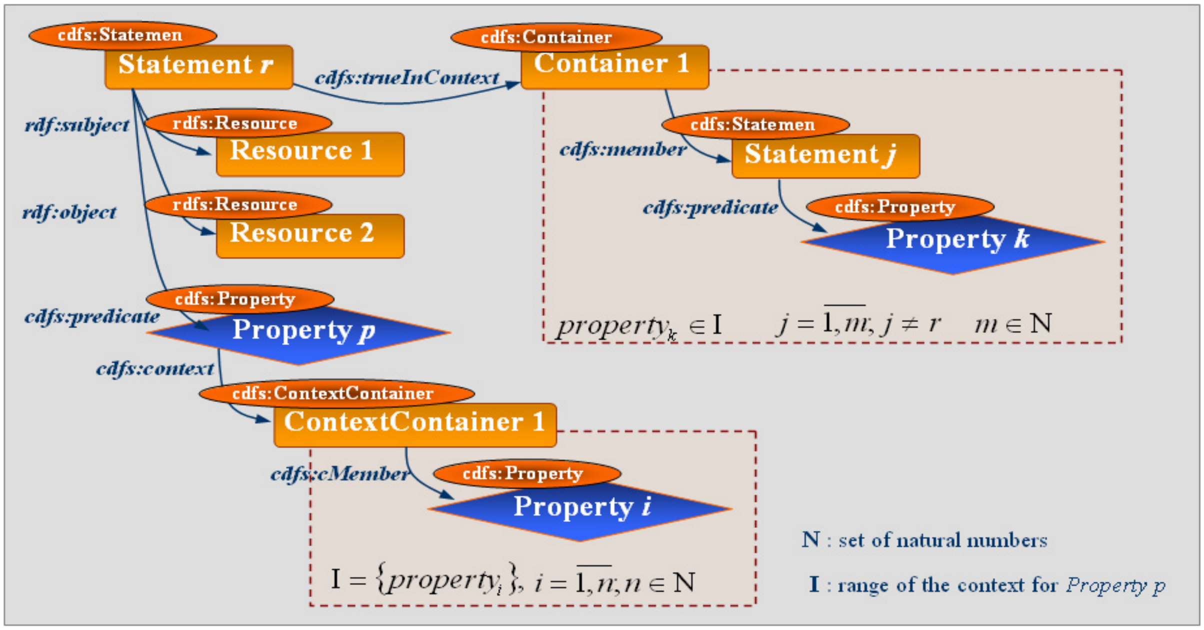


Figure 32 - Context tolerance range definition

Three rules correspond to the subproperty definition. In the same way as in the RDF specification, the domain and range classes of a subproperty should be the same classes or subclasses of the super-property domain and range classes. Additionally, the subproperty context (vector of the properties) should be covered by the context of the super-property. It means that each element of the subject property context vector (property) should be a subproperty of some super-property context vector element or a new property (is not presented in super-property context vector) (Figure 33).

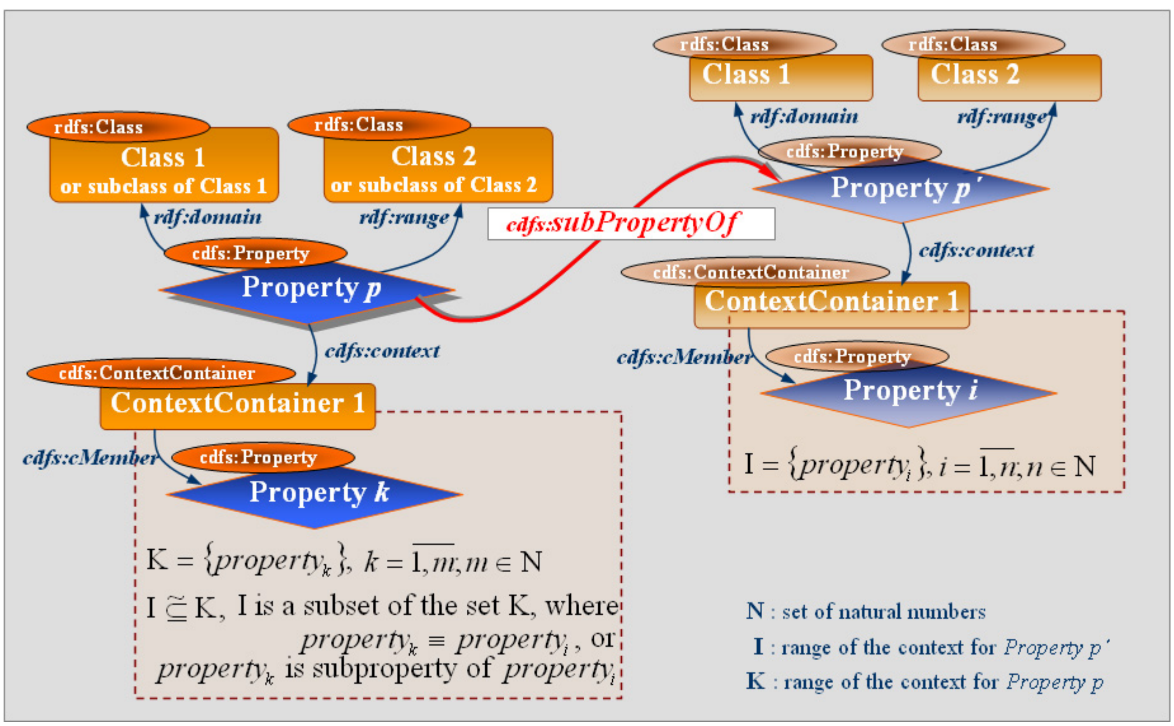


Figure 33 - Definition of the subproperty concept

CDF as a very general RDF extension provides us a tool towards more domain specific RDF extensions (RscDF, RgbDF and RpiDF) related to SmartResource and GUN concepts. It is known that RDF data model used to represent statements about resources. RscDF, RgbDF and RpiDF represent statements about "smart" resources (as a subset of resources). Main semantic peculiarity of RscDF is description of statements, which subject is RDF statement, predicate is either *true_in_context* or *false_in_context* and object is a container of RDF or RscDF statements (Kaykova *et al.* (2005b)). Main semantic peculiarity of RgbDF is description of statements, which subject is smart resource, predicate is one of behavioral properties, such as: *has_role, has_goals, has_behavior, executes*, etc and object is a container of RDF, RscDF or RgbDF statements (Kaykova *et al.* (2005c)). Main semantic peculiarity of RpiDF is description of statements, which subject is RDF, RscDF or RgbDF statement, predicate is one of rule/metarules/constrain properties, either *true_if*, or *false_if,* and object is a container of RDF, RscDF, RgbDF or RpiDF statements (Kaykova *et al.* (2006)).

The main ideas behind the three frameworks described above (General Adaptation Framework, General Proactivity Framework and General Networking Framework) and the appropriate conceptual difference between RscDF, RgbDF and RpiDF are shown in Figure 34.

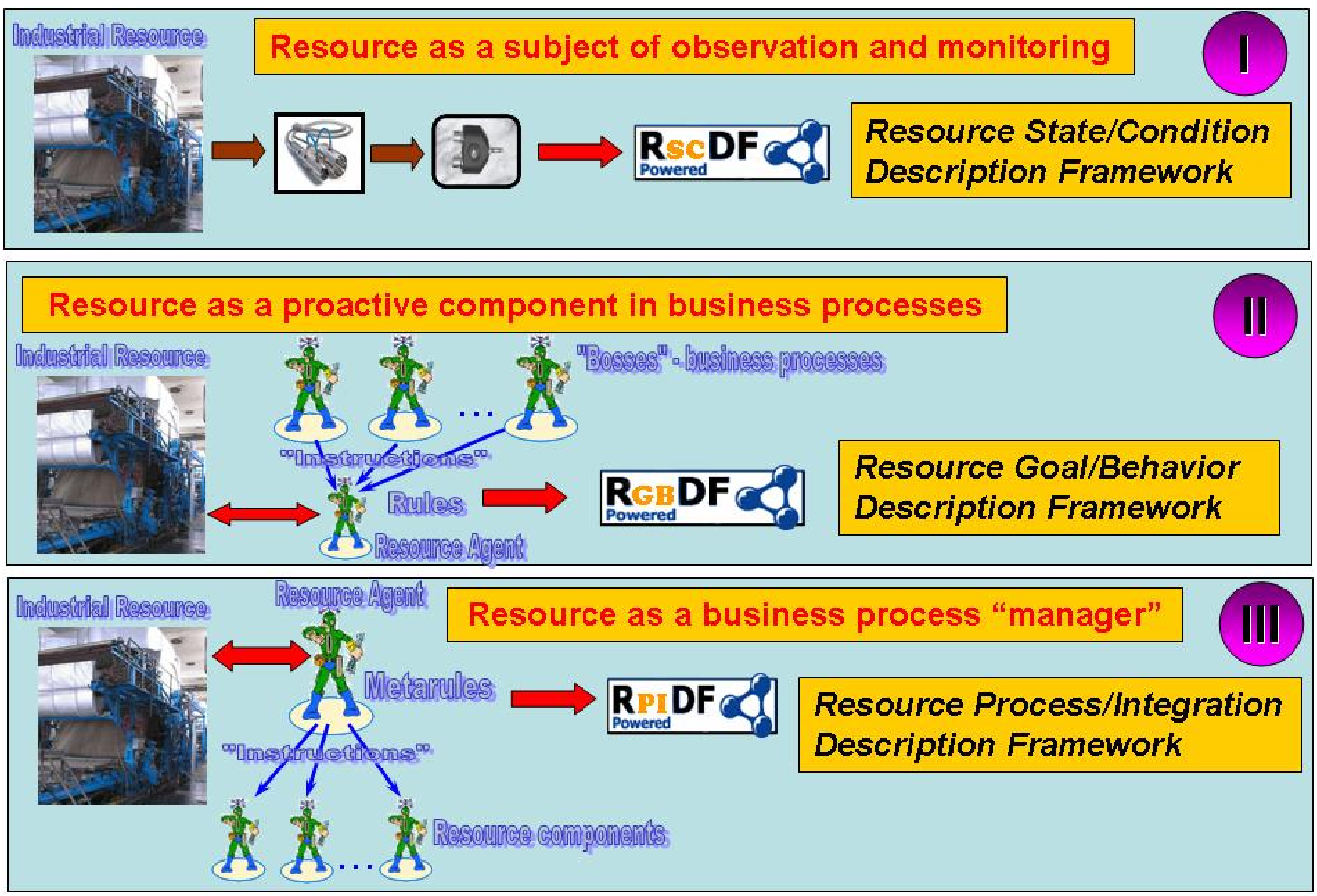


Figure 34 – The conceptual difference between three frameworks in SmartResource project

As it was mentioned above, the GUN environment meant for online condition monitoring and predictive maintenance of various industrial resources. Utilization of RscDF, RgbDF and RpiDF 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. The basic and more or less universal maintenance lifecycle of a resource (e.g. device, expert, service, etc.) and its contribution to the resource history is shown in Figure 35.

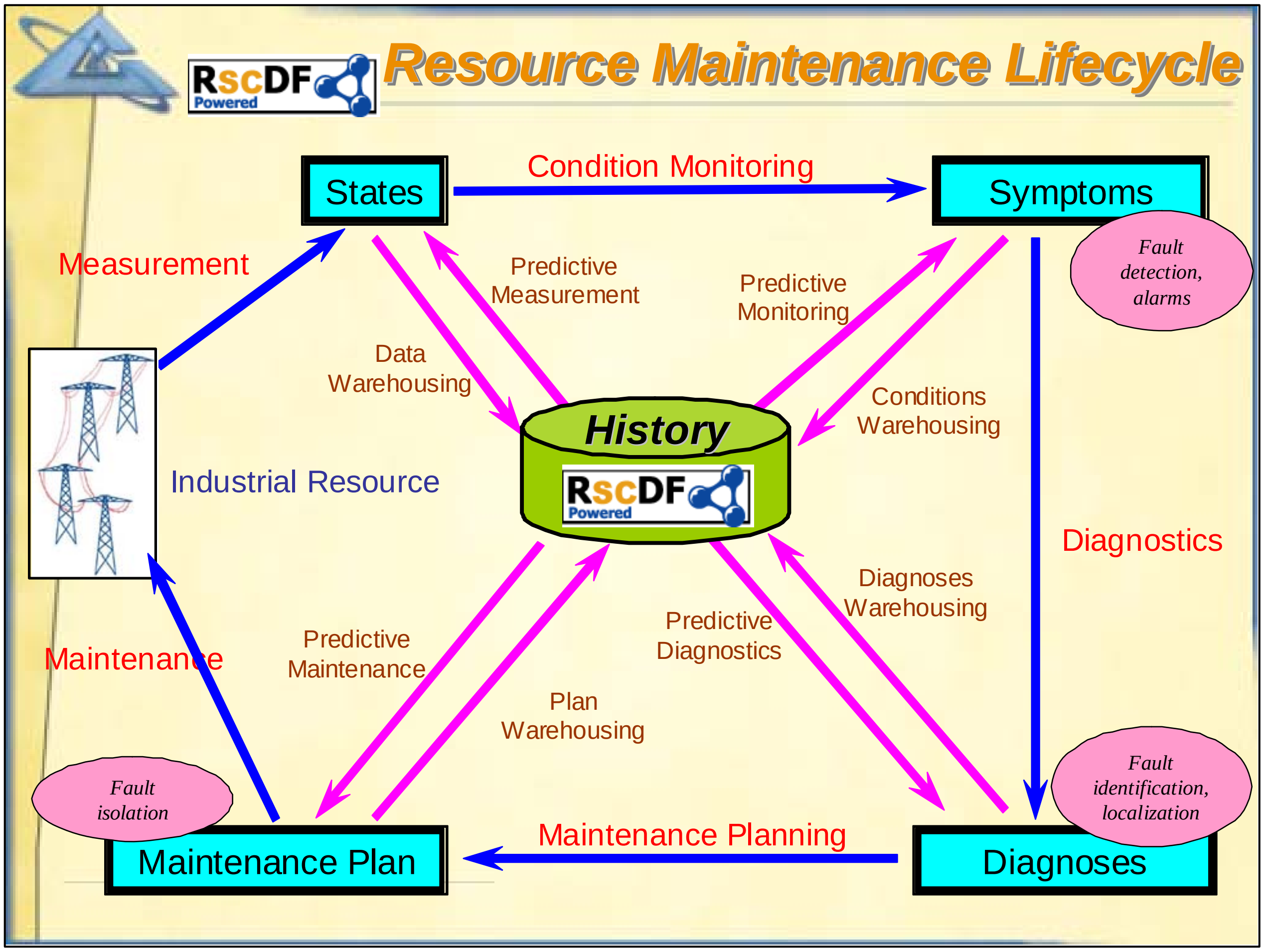


Figure 35 – Self-management lifecycle of a smart resource and lifeblog collecting

It is important to mention that such lifecycle has sense for really different types of resources including all GUN players: "service consumers", "service providers" and "decision makers". The terms "measurement", "condition monitoring", "diagnostics" and "maintenance", etc. have wider meaning than the traditional ones and can be applied to all tangible or intangible resources as the general concept of "self-management".

Collected history may be not only subject for querying, sharing, integration, etc, but also can provide useful patterns (discovered by data-mining tools), which can be used for predictive diagnostics, maintenance, etc. In Nikitin *et al.* (2005) we have shown how to apply RDQL query patterns to query such history storages, which are based on our RDF extensions.

8. SmartResource Platform Architecture

This section describes the recently achieved state of the GUN Platform for smart resources. The central to the platform is the architecture of a SmartResource agent depicted in Figure 36. It can be seen as consisting of three layers: reusable atomic behaviors (RAB), behavior models corresponding to different roles the agent plays, and the behavior engine.

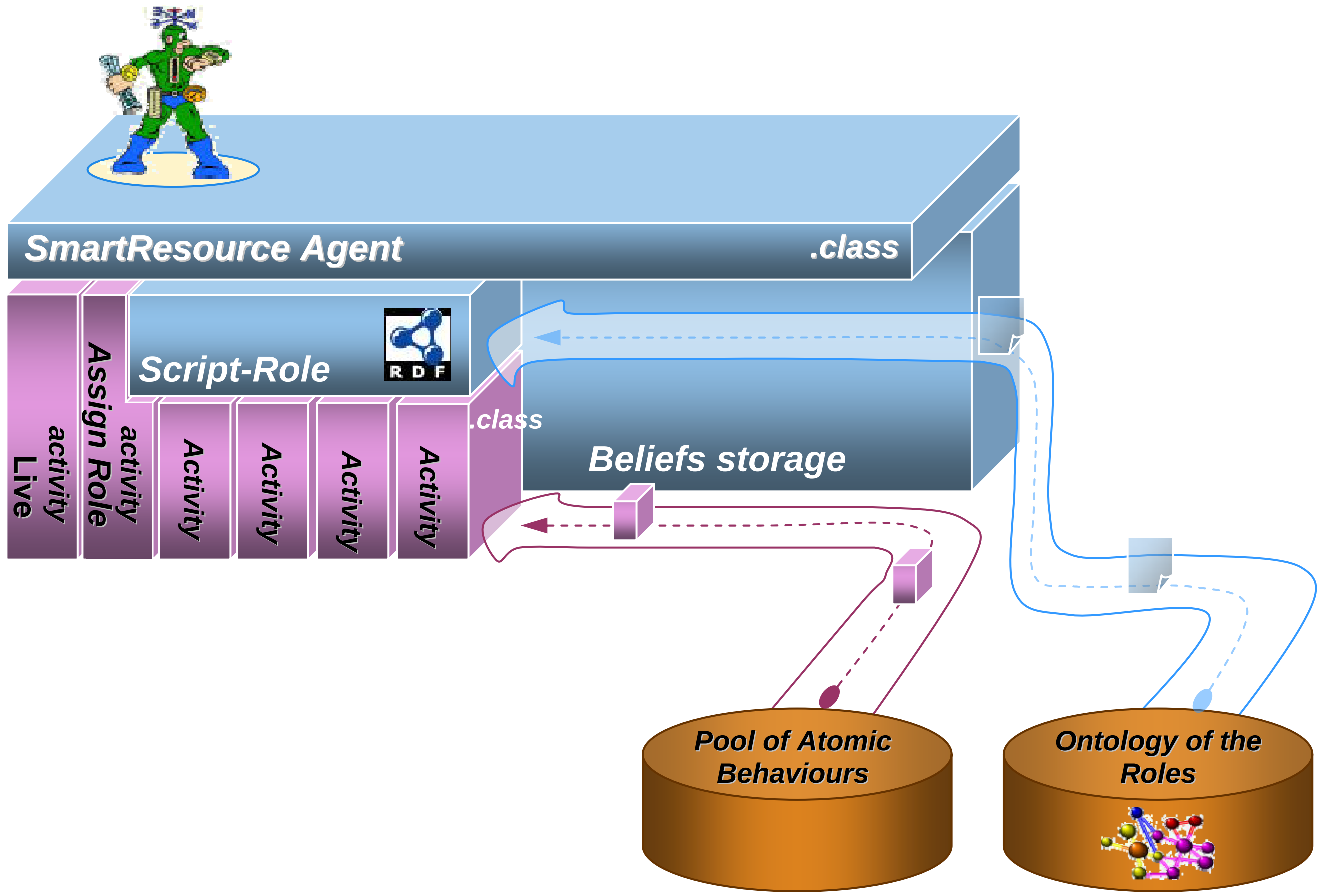


Figure 36 - SmartResource agent architecture

A *reusable atomic behavior (RAB)* is a piece of Java code implementing a reasonably atomic function. Therefore, RABs correspond to actuators and preceptors. As the name implies, RABs are assumed to be reusable across different applications, different agents, different roles and different interaction scenarios. Some examples of RABs from our case system (see the next section) are:

- RequestSenderBehavior and RequestReceiverBehavior – actuator and preceptor, correspondingly, related to sending and receiving requests for some service.
- DataSenderBehavior and DataReceiverBehavior – actuator and preceptor, correspondingly, related to sending and receiving the responses upon requests.
- ExternalApplicationStarterBehavior – popping-up an external application in order, e.g., to visualize some information to a human.

- DirectoryLookupBehavior – interacting with a system agent called directory facilitator (DF), which stores the mapping between agents and roles, in order to find agents playing a specific role.
- RandomSelectBehavior – randomly selecting an agent among several playing the same role.
- OneStepAuctionBuyerBehavior and OneStepAuctionSellerBehavior –implementing the buyer and the seller behaviors of the simplest auction: request for bids is sent, bids from all are received or time limit is expired, and the best bid is selected.

In the GUN Platform, the behavior of an agent is defined by the roles it plays in one or several organizations. Some examples of the possible roles: operator's agent, feeder agent, agent of the feeder N3056, fault localization service agent, ABB fault localization service agent, etc. Obviously, a general role can be played by several agents. On the other hand, one agent can (and usually does) play several roles.

A *behavior model* is an RgbDF document that is supposed to specify a certain organizational role, and, therefore, there is one-to-one relation between roles and behavior models. A behavior model consists of a set of beliefs representing the knowledge needed for playing the role and a set of behavior rules. Roughly speaking, a behavior rule specifies conditions of (and parameters for) execution of various RABs. Obviously, RABs need to be parameterizable. For example, RequestSenderBehavior takes such parameters as the agent to send the request to and the request itself. Notice that, in GUN Platform, if a behavior model specifies the need of interaction with another agent, that agent is always specified by its role, not name or another unique identifier of a particular agent. If several agents play the role needed, the behavior model is supposed to include some rules specifying a mechanism of resolving such a situation, e.g. random select, auction, etc. Different such mechanisms can of course be assigned to resolving conflicts with respect to different roles.

The *behavior engine* is the same for all the SmartResource agents (we mean that each agent has a copy of it). The behavior engine consists of the agent core and the two core activities that we named "assign role" and "live". The AssignRole activity is responsible for parsing RgbDF of a behavior model into the beliefs and behavior rules storages. The Live activity implements the run-time loop of an agent. Roughly speaking, it iterates through all the behavior rules, checks them against existing beliefs and goals, and executes RABs corresponding to the rules to be fired. Upon creation of the agent, the AssignRole activity needs to be invoked directly from the agent's core to parse the startup behavior model; however, all the later invocations of AssignRole (parsing of actual roles) are made according to that startup model. Therefore, AssignRole has the duality of being a part of the engine and a RAB in the same time.

As can be seen from Figure 36, in GUN Platform, agents access the behavior models from an external repository, which is assumed to be managed by the organization which "hires" the agents to enact those roles. It is done either upon startup of an agent, or if the organization requests an update to be made. Such externalization of behavior models has several advantages:

- Increased flexibility for control and coordination. Namely, the organization can remotely affect the behavior of the agents through modifying the behavior models. Another advantage is that the models can always be kept up-to-date.
- An agent may 'learn' how to play a new role in run-time; it does not need to be pre-programmed to do it.
- Inter-agent behavior awareness. How is discussed in the previous section, the agents not enacting a particular role can still make some use of the information encoded in its behavior

model. One reason is to understand how to interact with, or what to expect from, an agent playing that role.

As can also be seen from Figure 36, GUN Platform allows on-demand access even of RABs. If an agent plays a role, and that role prescribes it to execute an atomic behavior that the agent is missing, the agent can download it from the repository of the organization. In a sense, the organization is able to provide not only instructions what to do, but also the tools enabling doing that. The obvious additional advantages are:

- An agent may 'learn' new behaviors and so enact in a completely new role.
- Agents may have a "light start" with on-demand extension of functionality.

Technically, GUN Platform is implemented on the top of the Java Agent Development Environment (JADE). Therefore, the SmartResourceAgent is a subclass of jade.core.Agent. All the RABs have to be subclasses of some of the subclasses of jade.core.behaviours.Behaviour, e.g. OneShotBehaviour, CyclicBehaviour, and so on.

We plan to extend the platform functionality in near future as shown in Figure 37. As one can see this extension adds more flexibility due two more layers of declarative descriptions: policy constrains and configuration settings. By policy constrains we can model agent behavior as a member of several multi-agent organizations simultaneously. By configuration settings we can enable self-configurability of the platform in changing environment.

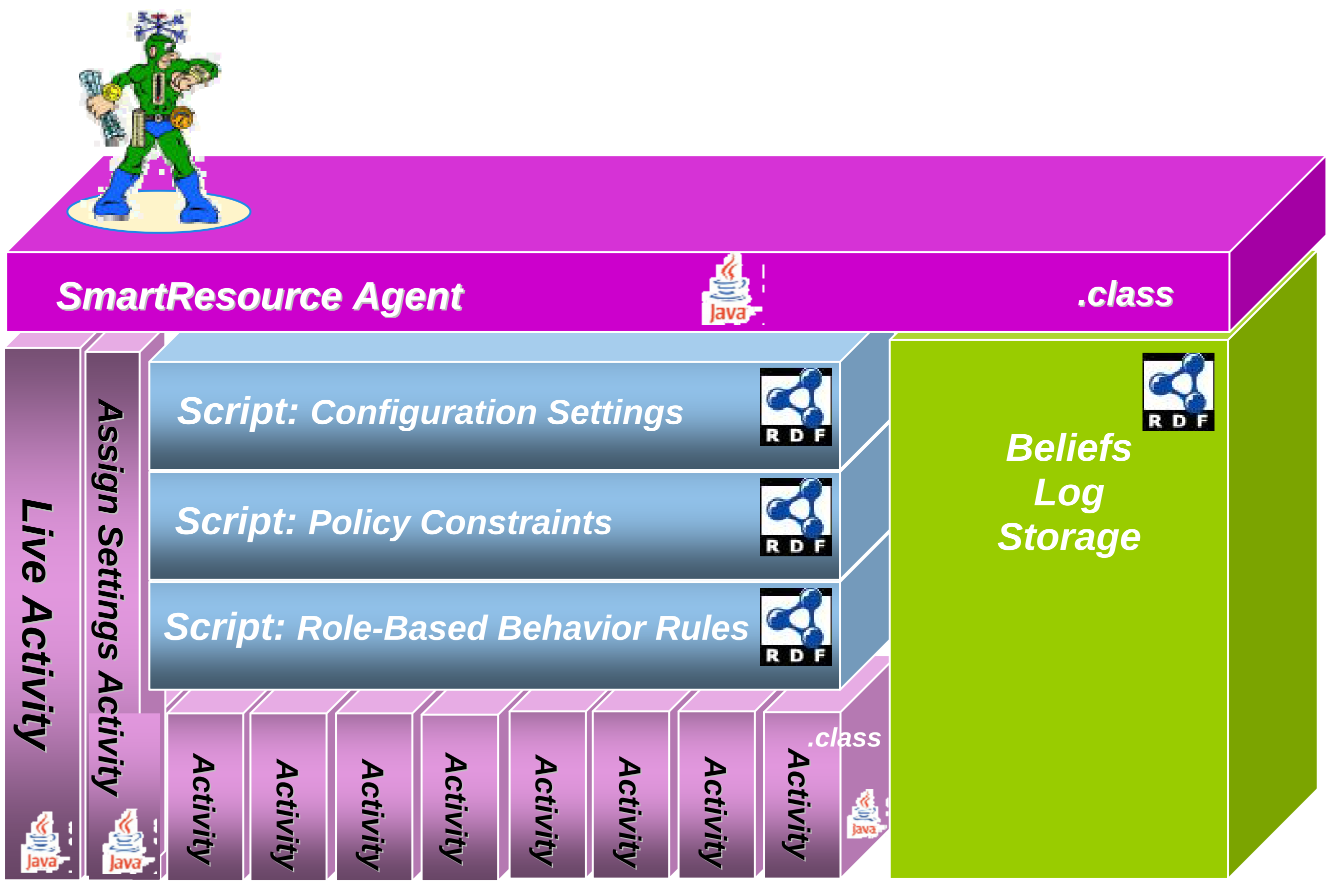


Figure 37 – Extended self-configurable agent platform architecture

More radical future extension of the platform can be related with making agent platform components to be agents themselves. In Figure 38, the 3-layered GUN platform for a particular industrial resource management is shown. The Agent Layer contains a resource agent who is responsible for a resource and also several GUN agents responsible for various software components needed for resource sensing, adaptation, condition monitoring, decision-making, maintenance, etc. Each of GUN agents is connected with appropriate software component from the Component Layer (e.g. resource sensor adapter, resource actuator adapter, alarm manager, etc.) and able to automatically invoke it whenever needed; or it can be connected to appropriate semantic storages at the Data Layer (which are: automatically annotated resource history, resource proactive behavior, or resource commitments with other resources). Data Layer components are linked to the GUN ontology (either distributed or centralized), which contains necessary reusable patterns for resource history, resource behavior and resource coordination. Each resource agent keeps record of the resource states and own mental states in RscDF format with link to industrial domain ontology. Each resource agent keeps set of needed behavior patterns according to its role in a business process in RgbDF format with link to GUN ontology. Each agent can keep needed adapters, histories, behavior sets, software components, commitments and reusable coordination patterns (in PpiDF) and other GUN resources on the own GUN agent-platform. On such platform, resource agent can communicate with other GUN resources agents locally. Shared ontology guarantees interoperability and understanding among resource agents. Industrial world will be represented in GUN environment with distributed history database, which can be queried by agents and is the subject of agent communication. All the components from the Component Layer and the Data Layer can be exchanged between GUN platforms, flexibly composed and reconfigured on-the-fly as result of context-driven agent coordination on the Agent Layer.

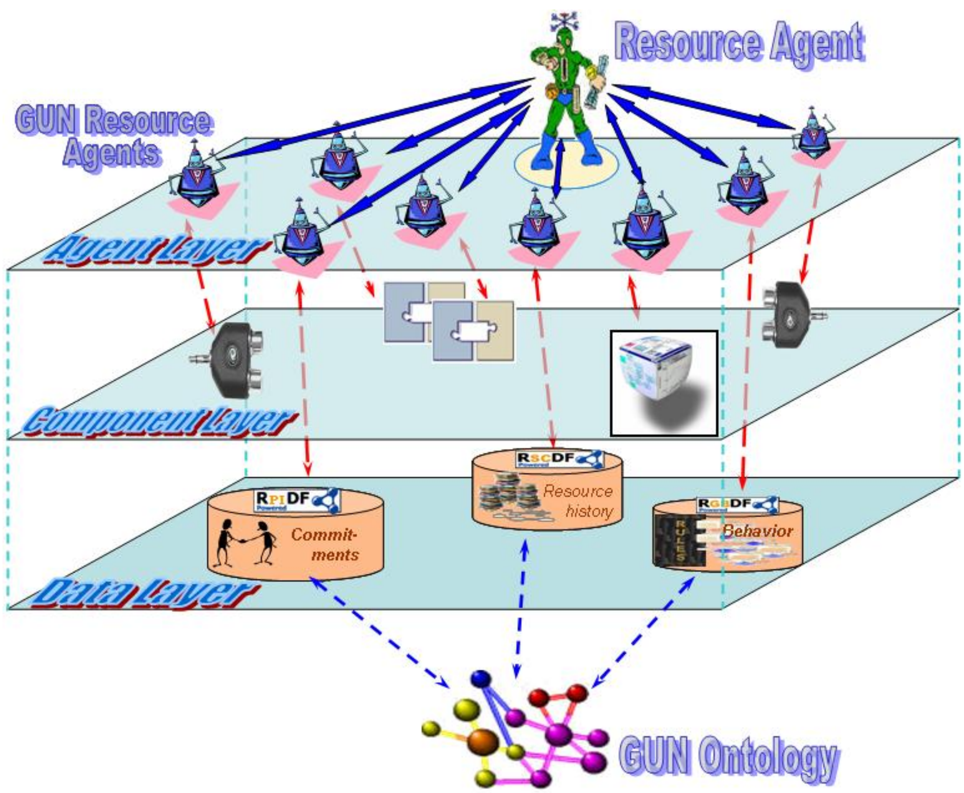


Figure 38 – A 3-layered GUN platform for managing smart industrial resource

9. SmartResource Demos and Industrial Case Studies

9.1. SmartResource Prototype Environment v. 1.0

For a practical testing of the developed General Adaptation approach, the first version of the target prototype environment has been implemented (Kaykova *et al.* (2005a)). The environment can be launched on one or several workstations, which meets the specified installation requirements. Figure 39 illustrates architecture of the implemented prototype environment.

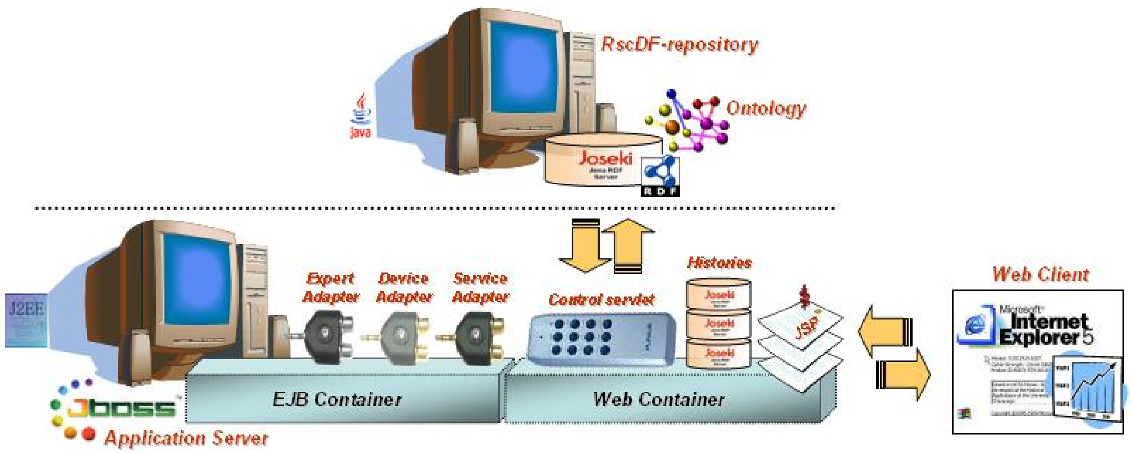


Figure 39 - Architecture of the SmartResource prototype environment, v. 1.0

For the process of software engineering the freeware and open source tools and technologies have been used. The whole environment is based on Java 2 Platform, Enterprise Edition[9] (J2EE) and was developed using Eclipse[10] Integrated Development Environment together with the Poseidon[11] UML-based modeling tool. Versioning control was carried out with help of the CVS[12] 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[13] RDF server. For creation of a local history cache, Jena[14] 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[15]. Visualization of the internal processes of the prototype environment was organized using a set of Java Server Pages[16] (JSP). Demonstrations were carried out using Internet Explorer web browser.

---

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

Web Service adapter incorporated a simple sample of learning algorithm (KNN-method) wrapped by a web service container using Axis[17] and Lomboz[18] (see Figure 40). The adapter using 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.

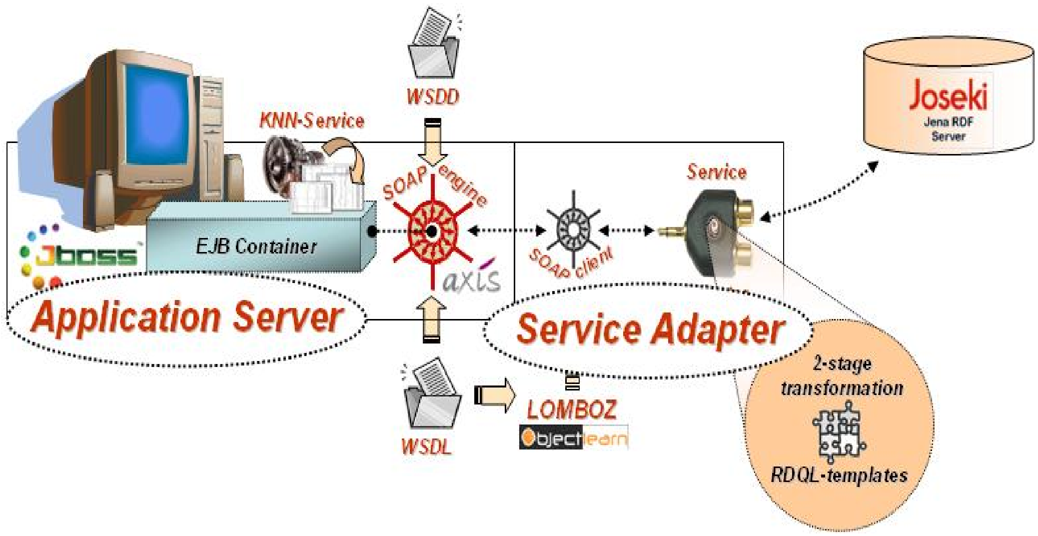


Figure 40 - Implementation architecture of the Web Service adapter

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 41). Involvement of the JFreeChart[19] open Java library allowed generating images for representation of the device states. Human Expert is requested for a diagnostics via e-mail.

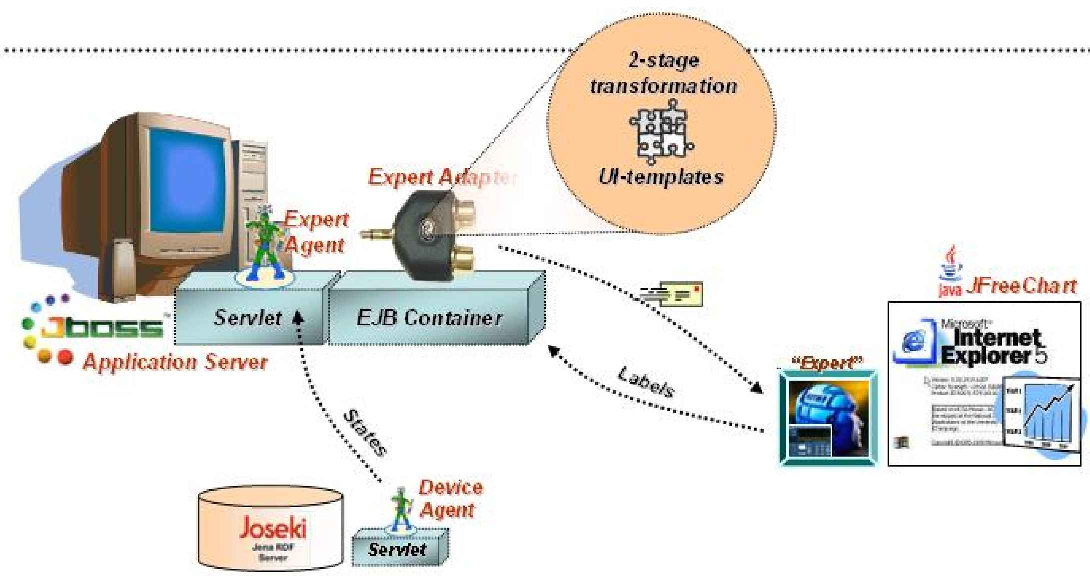


Figure 41 - Implementation architecture of the human expert adapter

[17] http://ws.apache.org/axis/, official webpage of Axis Apache
[18] http://www.objectlearn.com/index.jsp, Lomboz ObjectLearn Eclipse plugin
[19] http://www.jfree.org/jfreechart/, JFreeChart – free Java class library for generating charts

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 42). 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.

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. 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.
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.

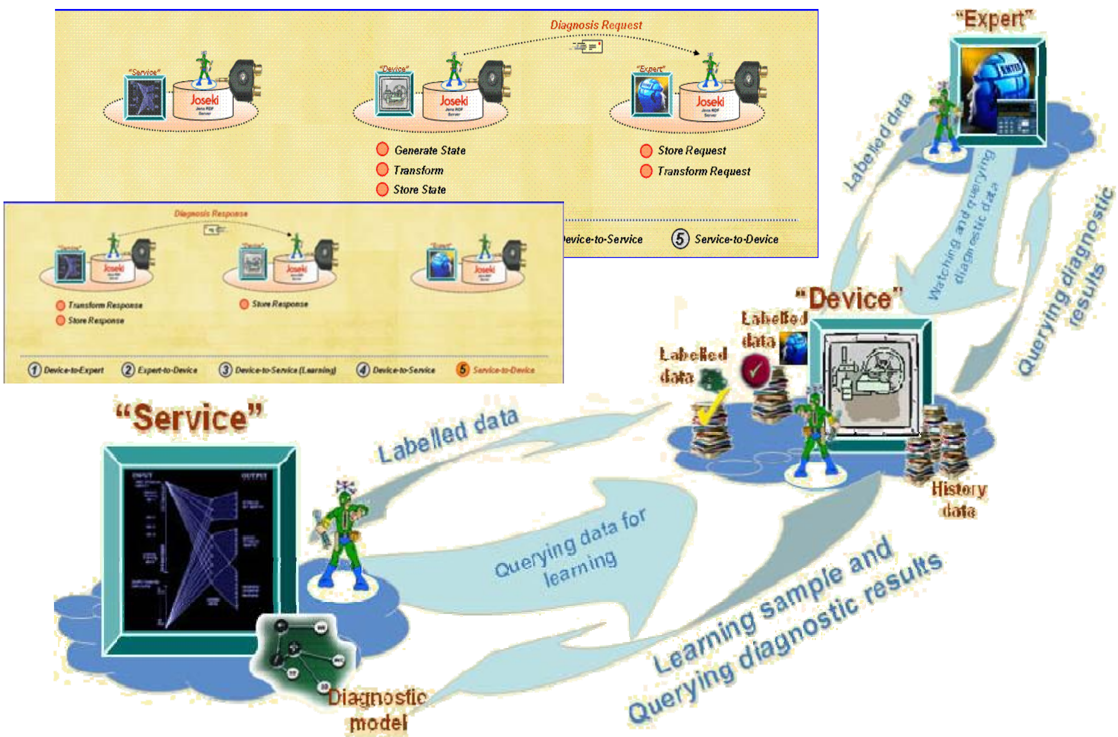
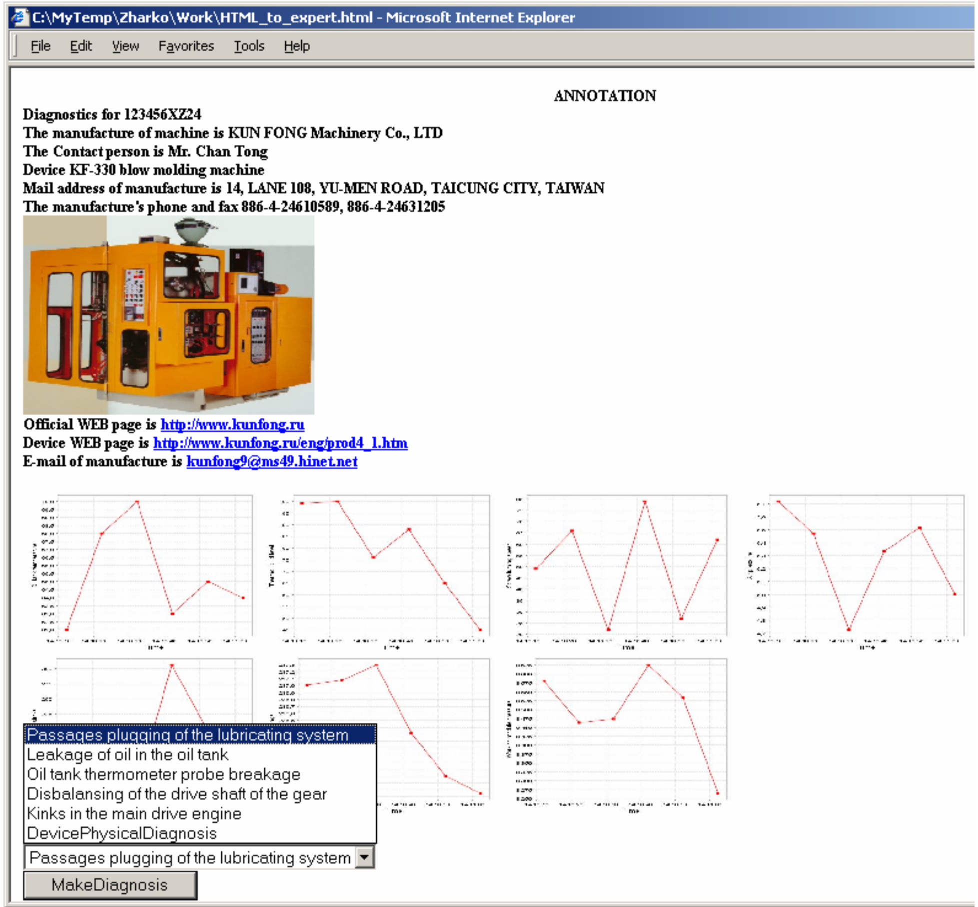
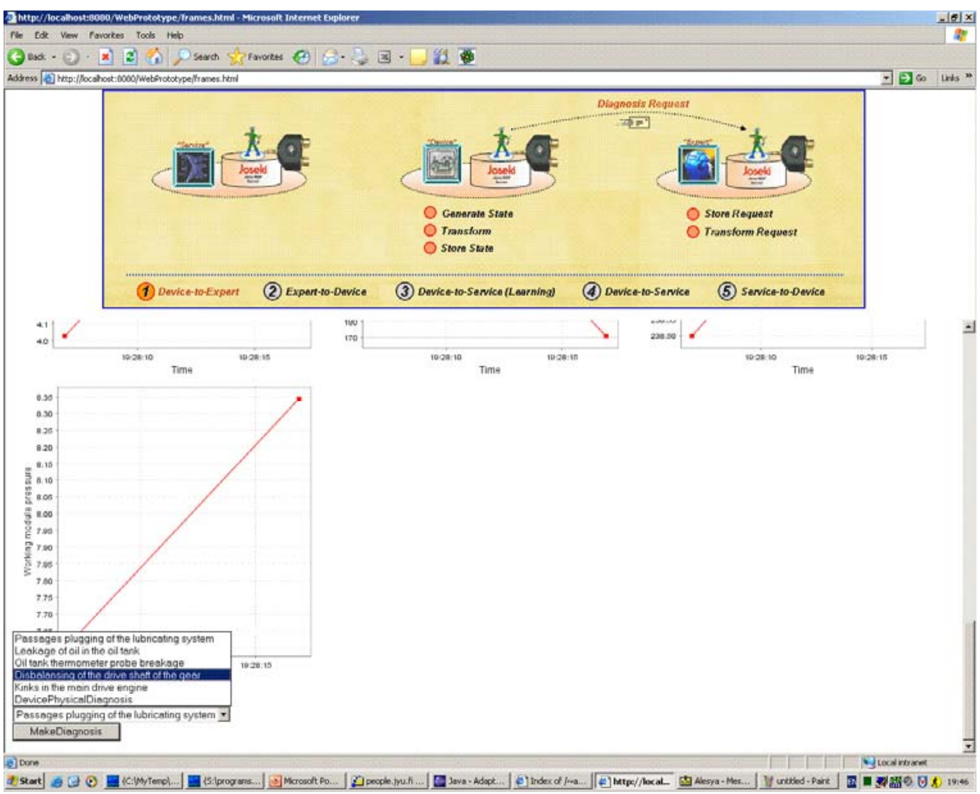


Figure 42 – Use case scenario of the SmartResource prototype v. 1.0

The screenshots of SmartResource prototype environment, particularly stage of expert diagnostics of the device, are shown in Figures 43-44. Here one can see hat the expert can observe few last values of the device parameters and can select one of possible diagnoses from the list automatically created from the ontology.





Figures 43-44 – Screemshots of the SmartResource prototype environment v.1.0

9.2. SmartResource Prototype Environment v. 2.0

The scenario of interactions between a Device, an Expert and a Web Service that was implemented in the SmartResource prototype environment v. 1.0 has been automated towards v. 2.0 (Naumenko et al. (2005)). The logic of interactions has been implemented as a multi-agent system involving DeviceAgent, ExpertAgent and WebServiceAgent respectively (see Figure 45).

As a basis for implementation of the interaction scenario between SmartResource agents the Java Agent Development Framework (JADE) has been chosen. Such choice has been made, because Java language is the basis for JADE that makes its integration with previous version of the SmartResource Prototype Environment easy. Additionally, the JADE platform is mature in providing a variety of tools for the debugging and deployment phases of the agents. JADE fully follows FIPA standardized specifications that is important for further ontological description of multi-agent coordination. In general, the implementation task assumes migration of the scenario's logics from the Control Servlet to the community of agents implemented in JADE. On the other hand, adapters that were implemented during previous project year, reside at the JBoss Application Server as they are. It is one of the challenges to implement the access of agents hosted by JADE to the adapters. So far, developers of JADE have provided a possibility to implement behaviors of agents using the hierarchy of specific classes. This structured approach to modeling behaviors makes JADE platform even more suitable for experimental research of the RgbDF schema and RgbDF engine.

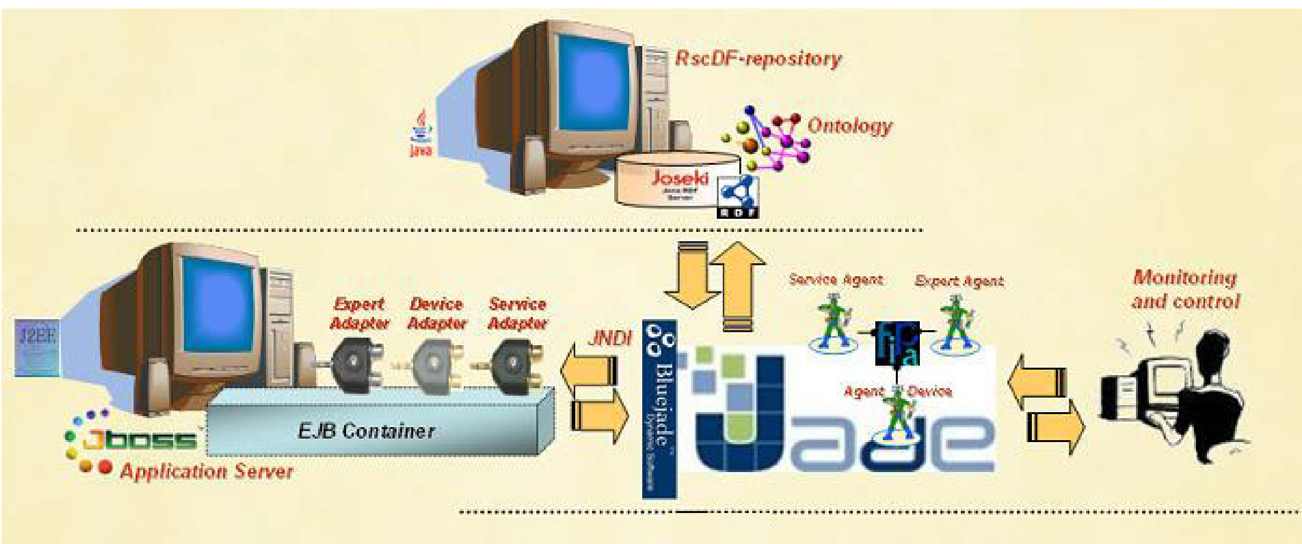


Figure 45 - Architecture of the SmartResource prototype environment, v. 2.0

The implemented agents (for device, service and expert) access the adapters for data transformation needs. For this purpose, an abstract class *ResourceAgent* has been designed. It implements the initialization of local history storage of an agent from common history stored at the Joseki server. Additionally the class makes necessary preparations for a successful lookup of the adapters by agents: an instance of a context (JNDI naming directory) that allows for adapters (implemented as EJBs) to be found by their names.

In order to succeed with the interaction of the agents on the platform, we have to start all the platform components in a predefined order. First the Joseki server must be started because every agent initializes an appropriate adapter, and requests for ontology from joseki storage. Next we start JADE platform as such without agents on it. As far as resource adapters are implemented mainly as EJB's, we start the JBoss server with adapters. During the JBoss initialization, an ExpertAdapterAgent (helper agent, which is a part of expert adapter) is created and deployed to JADE. Then we deploy Service and Expert agents, which are ready to accept incoming request messages. Now the AgentDeviceGenerator and DeviceAgent can be started. These two agents constitute the initial point of the platform operation, as far as they originally generate messages, which go to expert and service agents. When all the agents are running, the configuration of JADE is as it is shown in Figure 46.

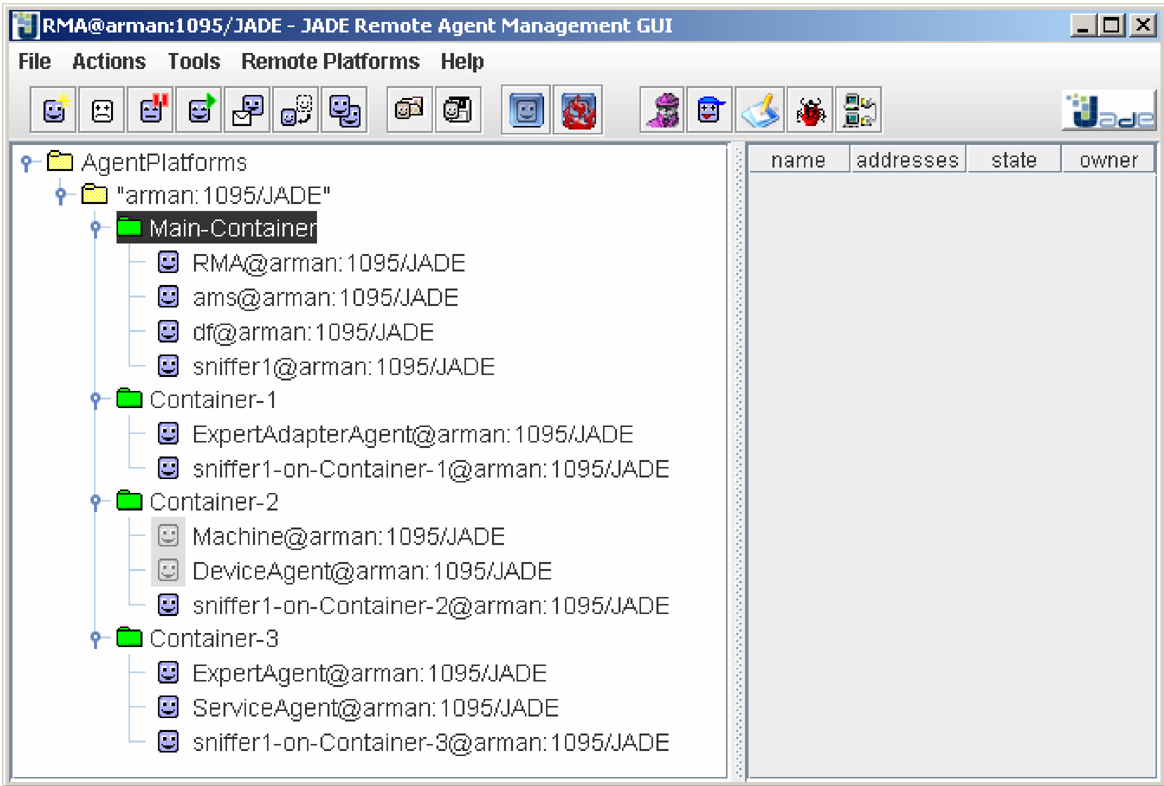


Figure 46 – General view of the JADE platform

According to the modeled scenario, the DeviceAgent receives messages with states in XML format from AgentDeviceGenerator, uses DeviceAdapter implemented earlier for transformation of the XML message into RscDF representation. The state in the RscDF format is stored in the local history of the agent and further every minute the DeviceAgent generates alarm message to ExpertAgent (see Figure 47).

The behavior of the ExpertAgent comprises the receiving message with request for diagnostics from DeviceAgent, then transforming this message to HTML using ExpertAdapter, getting expert's response, transforming to RscDF and sending it back to DeviceAgent. As it was implemented previously, the ExpertAgent sends a request for diagnostics to human expert via e-mail service, which contains link to a diagnostics page. This page is generated on the fly and is published on a web server. When expert opens the diagnostic page and makes a diagnosis, the data of the form is sent to a servlet, which in turn invokes an ExpertAdapterAgent's method to send a message to an ExpertAgent. When ExpertAgent receives a diagnosis from ExpertAdapterAgent, it transforms it into RscDF using the ExpertAdapter and sends the RscDF to a DeviceAgent (see Figure 48).

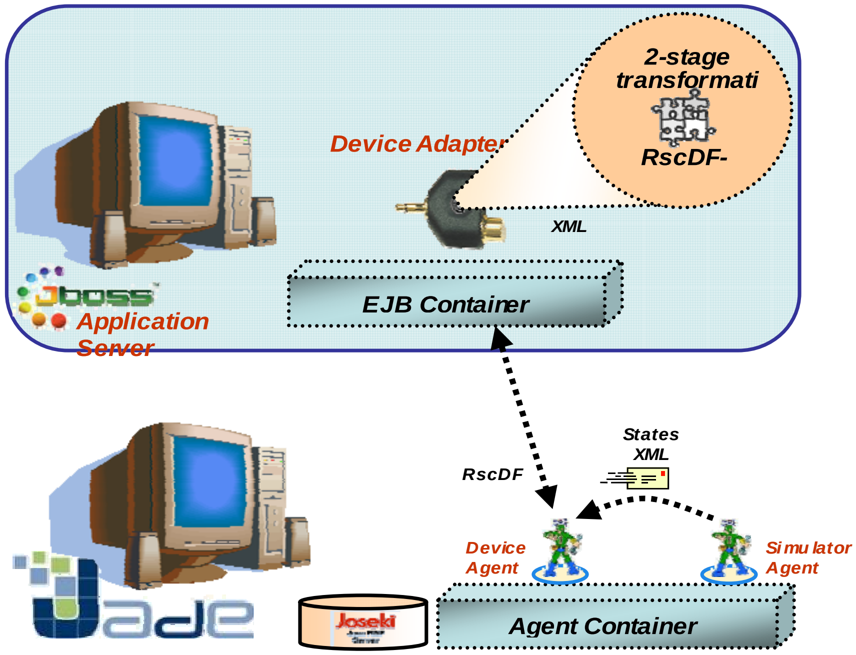


Figure 47 – Behavior of the DeviceAgent on JADE platform

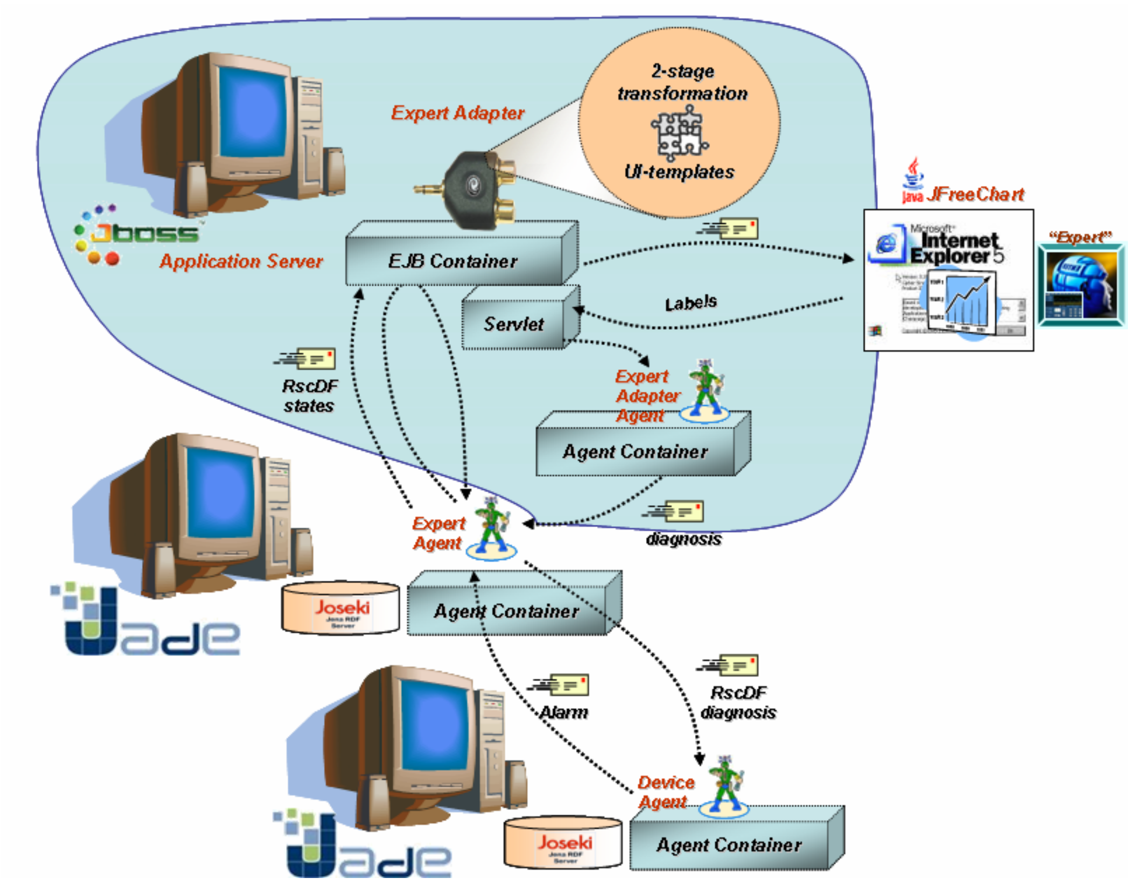


Figure 48 – Behavior of the ExpertAgent on JADE platform

The ServiceAgent logic is quite simple. It implements one cyclic behavior for accepting messages from the device. The messages can be of two types – request for diagnostics or request for learning. In case of learning request, the adapter performs transformation and then the behavior is added, which sends a confirmation to a DeviceAgent. When the diagnostic request is received, it is processed in two transformation steps. On the first step the classification as such is done, which returns diagnosis URI. On the second stage, the diagnosis URI is wrapped into an RscDF message using appropriate template. After the second transformation step, the OneShotBehavior is added, which sends the response with diagnosis to a DeviceAgent (see Figure 49).

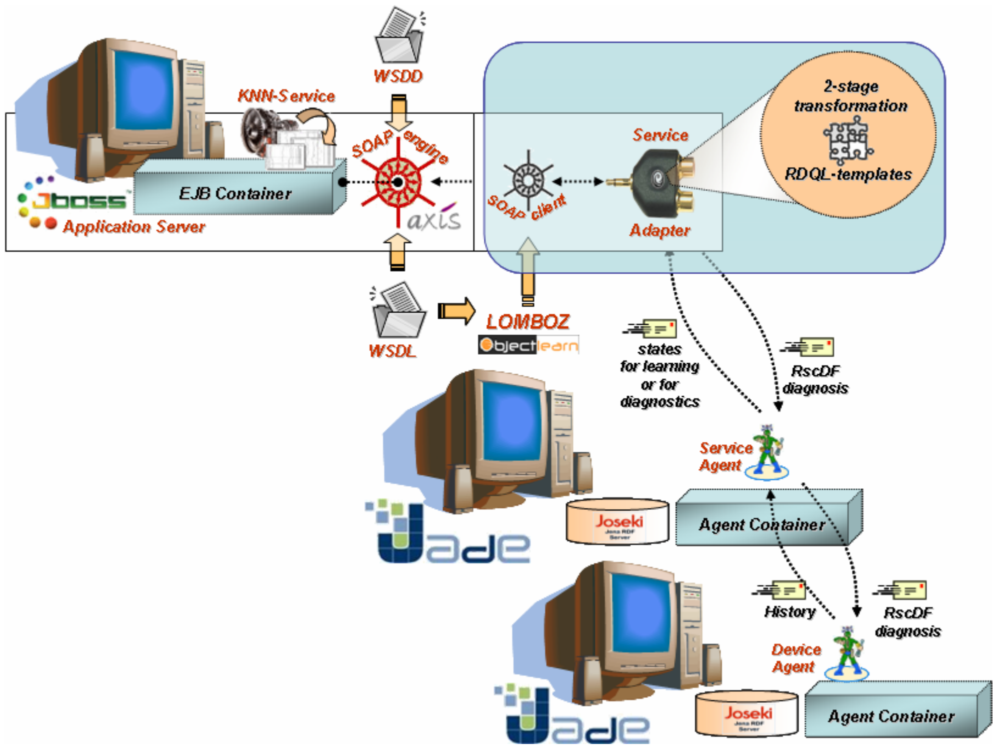


Figure 49 – Behavior of the ServiceAgent on JADE platform

The platform in runtime performs in asynchronous mode. The messages flow can easily be reconfigured if we want to change the logic of some of the components. In current implementation, the request for diagnostics is sent to an expert until enough learning examples for service are collected. After the service has learned, all the diagnostic requests are sent to ServiceAgent. The logic can be changed e.g. that every second example goes to Expert. This kind of configuration may be reasonable, when the service is not reliable enough. Another possibility is to send the diagnostic request to both Expert and Service. This kind of logic can be implemented for Service quality verification. Thanks to the fact, that the platform consists of highly independent agent entities and unified semantic interchange format, the modification of logic becomes fairly simple and converges to modification of one java method in the DeviceAgent.

The transformation of the SmartResource prototype environment v1.0 to v2.0 was aimed at automating the platform in terms of agent technology, which would act as a set of agents representing resources behind them. Supplying every resource by its own agent, we have explored the implementation specifics of the asynchronous message exchange applied to the use case of the knowledge transfer from expert to service. The implementation has discovered a new type of interaction with the resources which have undetermined response time and web-based interface. An expert as the most complex resource for adaptation required a lot of efforts to be done towards weaving different technologies into logically bundled component. The complex interoperation tasks between JADE and application server included EJB invocation from an agent platform, on-the-fly creation of the html data, dynamic processing of the expert response and artificial bridging from the web server to an agent platform via creation and posting an agent from the servlet to the running platform.

9.3. Metso Automation case: IT infrastructure in paper industry

This case study (Nikitin *et al.* (2007)) is related to SmartResource partnership with Metso Automation. The company is a global supplier of process industry machinery and systems as well as know-how and aftermarket services. The company's core businesses are fibre and paper technology, rock and minerals processing, and automation and control technology. Metso's strategy is based on an in-depth knowledge of its customers' core processes, close integration of automation and ICT, and a large installed base of machines and equipment. Metso's goal is to transform into a long-term partner for customers. Based on the remote service infrastructure, it develops solutions and services to improve efficiency, usability and quality of customers' production processes through their entire life cycles.

9.3.1. Remote Service Infrastructure. Metso's remote service infrastructure consists of service provider's Central Hub and several customers' Site Hubs, which are integrated together over the network. Site Hub solution is based on an EAI Platform (see Figure 50).

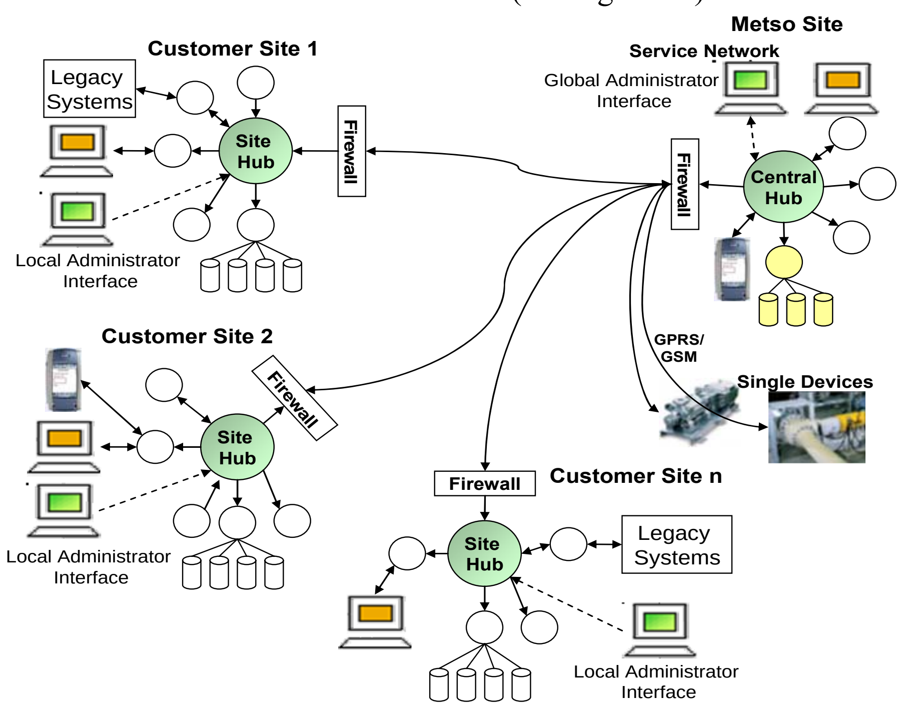


Figure 50 – Metso SiteHub network architecture

The key issues in Site Hub solution are: open standards, information security, reliability, connectivity and manageability. These requirements are met by combining a traditional EAI platform with new features, which are specially designed upon industrial needs. Hub-based integrated infrastructure combined with secure connectivity allows easy incorporation of new business logic on both customer and Metso sites. Messaging mechanism between customers and Metso provides very flexible medium for information exchange and new service provisioning.

9.3.2. Logging and Annotation of Maintenance Data. The main purpose of the SmartResource application based on above architecture is to semantically store fault data, coming to the Maintenance

Center from client paper machines. Based on web service messaging infrastructure, Metso has established a flow of SOAP[20]/XML messages with the maintenance data from Central Hub to a dedicated server at the university network. The messages are aligned to a specified XML format and pass from the Site Hubs to a Central Hub, which redirects them to the web application on the university site. The system we have developed can be divided into two main subcomponents – Message Handler and Message Browser (see Figure 51).

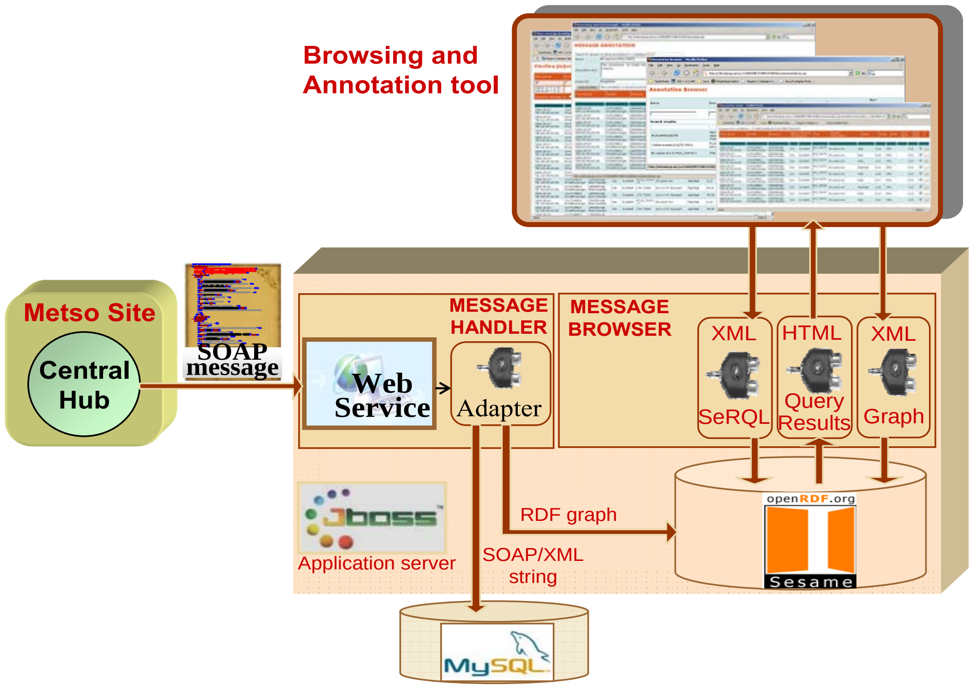


Figure 51 - Architecture of the system

Message Handler receives and processes SOAP messages from customers. The entry point of the component is a *MessageReceiver* class wrapped as a web service, which processes the SOAP message and takes the XML content from the header and body of it. Then it invokes the adapter to transform the XML content to the RDF[21]-graph object and store it to the Sesame[22] RDF storage. The original SOAP message is stored as a string to the MySQL database.

Adapter is one of the key enabling elements of the platform. It produces RDF-graph, compliant with the ontology. The transformation logic of the adapter generates a backbone of the RDF-graph. Then the XML-content is processed and transformed to the RDF statements, which are then added to the RDF-backbone.

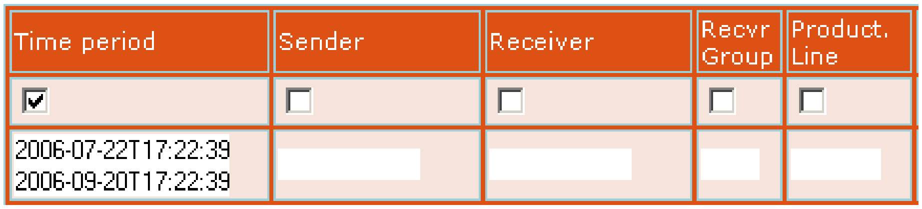
Ontology plays a role of a schema for all data within the storage. On a domain analysis stage, we have distinguished main concepts and linked them with the needed relationships (properties in OWL (OWL) and RDF terminology).

Message Browser component provides a web based interface for browsing and filtering messages stored in RDF-storage according to user-defined filtering criteria. The purpose of message filtering is to distinguish the groups of messages leading to exceptional situations. Using a filtering mechanism, the domain expert can distinguish message groups and annotate them. The annotation is stored to the RDF-storage and can be used as a sample for machine learning algorithms.

[20] SOAP – Simple Object Access Protocol, http://www.w3.org/TR/soap/
[21] RDF – Resource Description Framework, http://www.w3.org/RDF/
[22] Sesame - An RDF Schema-based Repository, http://www.openrdf.org/

The client-server interaction is implemented using AJAX[23] technology, which allows more dynamic script-based interaction with the server.

When a user performs any action which requires invocation of server functionality, the script on a client side wraps required parameters as an XML document and sends it in a request over HTTP to the server. For example, in order to filter the messages, user selects needed parameters and specifies parameter values in the corresponding textboxes (see Figure 52).



Time period	Sender	Receiver	Recvr Group	Product. Line
☑	☐	☐	☐	☐
2006-07-22T17:22:39 2006-09-20T17:22:39				

Figure 52 - Search parameter specification

By pressing the "query" button, the selected filtering parameters are wrapped into the XML document. On a server side *Control Servlet* processes the HTTP request and distinguishes what action to take, depending on a root tag of the XML document. When the query is executed, *Message Browser* sends a response to the client. On the client side a dedicated callback script function is waiting for a reply from the server. It processes the response and shows the result in a web browser.

9.3.3 Integration with the agent platform. In order to reconsider places and roles of the system components in the SmartResource agent-based environment, we have introduced the following communication scenario (see Figure 53).

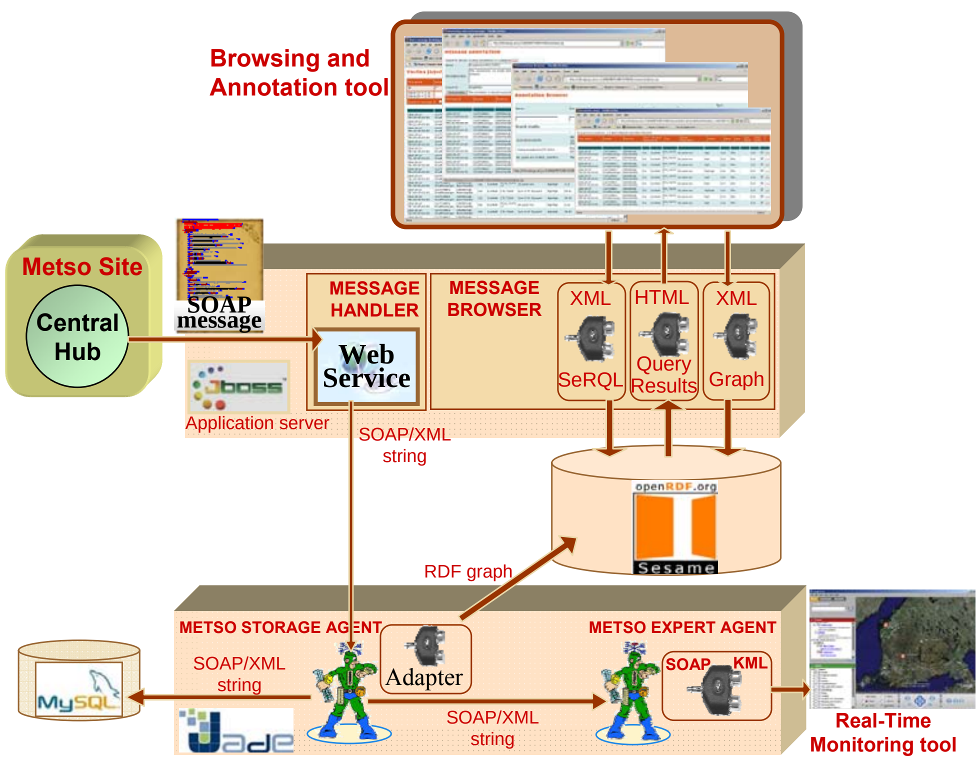


Figure 53 - Agent-enabled architecture

[23] AJAX - Asynchronous JavaScript and XML – a web development technique for creating interactive web applications

We have assigned an agent to manage RDF-storage activities (*Metso Storage Agent*) and provided a *Metso Expert Agent* to interact with a maintenance expert. The messages coming from customers are forwarded to *Metso Storage Agent*, which incorporates *Adapter* to perform transformation and storage. Then, *Metso Storage Agent* forwards the message to *Metso Expert Agent*, which updates the situation on a Real-time Monitoring Tool and provides an expert with the message content and a link to the browsing and annotation tool. By clicking the link the expert opens browsing and annotation tool with the pre-selected message. The expert now can specify query based on the message parameter values.

This prototype presents a solution for logging and annotation of real-time data in paper industry maintenance domain utilizing semantic web tools. The system is built on top of the web service-based infrastructure of the industrial partner. The main quantitatively differentiating features of the system are: integral data storage mechanism (RDF-based), easy-to-extend model (ontology), simple and dynamic querying mechanism, and application of data adaptation technique. Next, we have made a pilot integration of system elements (smart resources) with the JADE agent platform and implemented a simple scenario of agent to agent communication.

The above case study demonstrated that data integration problems are easier to solve with semantics-based solutions, although, the responsibility of the system analyst and architect is much higher. The evolution of semantic storages continues and we expect that new versions will incorporate smart "full-proof" features to prevent actions leading to inconsistency of data. Ontology storages bring clear differentiation of the data as such, and applications, which consume this data. The data is not just stored in a separate subsystem like RDBMS, but it is rather logically independent from any application, which accesses it. To manage more independent entities of the prototype we have introduced agent-based communication and coordination. In the next versions of the prototype we plan to make agents self-aware and robust to changes. Agents will resolve inconsistency problems by negotiation and self-configuration algorithms.

9.4. ABB case: distributed power network maintenance

This section describes a case study (Terziyan & Katasonov (2007), Salmenjoki *et al.* (2007)) in the domain of distributed power network maintenance we have been performing, starting from early 2006, in collaboration with ABB (Distribution Automation). The goal is to study the potential add-value which ABB could receive from introducing Semantic Web technologies and GUN framework in particular into their business. Development of a prototype, for demonstration of the concept purposes, was a part of the study as well.

9.4.1. The Vision. A very brief description of the domain follows. A basic unit of monitoring in a power network is a *feeder*, which is a section of the power line including all the poles, conductors, insulators, etc. The start and the end point of a feeder are *substations*, whose task is to transform the electric power e.g. from high-voltage to medium-voltage or from medium-voltage to low-voltage. In addition to the transformer, any substation naturally includes the devices monitoring and protecting both the incoming and the outgoing feeders. Such *protection relays* automatically monitor the state of the feeder in terms of voltages and currents, are able to disconnect the feeder if a significant *disturbance* is registered, and to automatically re-close the circuit after a specified time (and to break it again if the disturbance persists).

Persistent disturbance is usually a sign of a *fault* in the network, which could be e.g. earth fault (conductor falling of the ground), short-circuit (could be caused e.g. by a tree falling on a line with bare conductors), or open circuit (broken line). Restoration of the network, after a fault occurs, includes *fault detection*, *fault localization* (estimating the geographic location of the fault), and of course fault removal. In meanwhile, network reconfiguration may also be performed, with a goal of e.g. minimizing the number of customers who will suffer outage of power until the fault is removed.

As mentioned, the fault detection is performed by protection relays. The rest is performed in the *operation centers* with participation of human *operators*. In case of a fault, protection relay sends an alarm to the operation center and also sends a dataset with recorded disturbance: several-second history of all the monitored parameters with a high frequency of sampling (0.5 ms or so). A certain operation center controls a sub-network of the integral power network. The operators use systems, which belong to the MicroSCADA Pro product family, like DMS 600 or MicroSCADA Pro Distribution Management System and SYS 600, which is MicroSCADA Pro Control System. These systems provide an integrated graphical view over the sub-network, provide data acquisition from the substations and remote control over the relays, switches, etc. The systems like DMS also include implementations of various algorithms: for fault localization, for calculation of optimal reconfiguration of the network and other.

ABB is a vendor of hardware and software for power networks. The medium-voltage sub-networks are owned, controlled and maintained then by some local companies, e.g. Jyväskylän Energia for the city of Jyväskylä, and Vattenfall for all the rural areas around. It is noticeable that the operation centers of different companies have no connection to each other, so information exchange among them is nearly impossible. In the case of a fault affecting two different sub-networks, such information exchange, though, may be very important, for all of fault localization, network reconfiguration, and network restoration. Introducing an inter-organizational GUN system could solve this issue (Figure 54). The information flow will go through the agents representing the sub-networks on the GUN platform. Utilization of Semantic Web technologies will allow such interoperability even if the sub-networks use software systems from different vendors (ABB is not the only one), and thus maybe different data formats and protocols.

Figure 54 - Scenario: sub-networks interoperability

The second scenario in our vision is related to *a new business model* that ABB could implement. At present, all ABB expertise gets embedded into hardware or software systems and sold to the customers as it is. A new business model would be to start own Web-service providing implementation of certain algorithms, so the ABB customers will utilize those algorithms online when needed (Figure 55). ABB will be always able to update algorithms, add new ones, and so on.

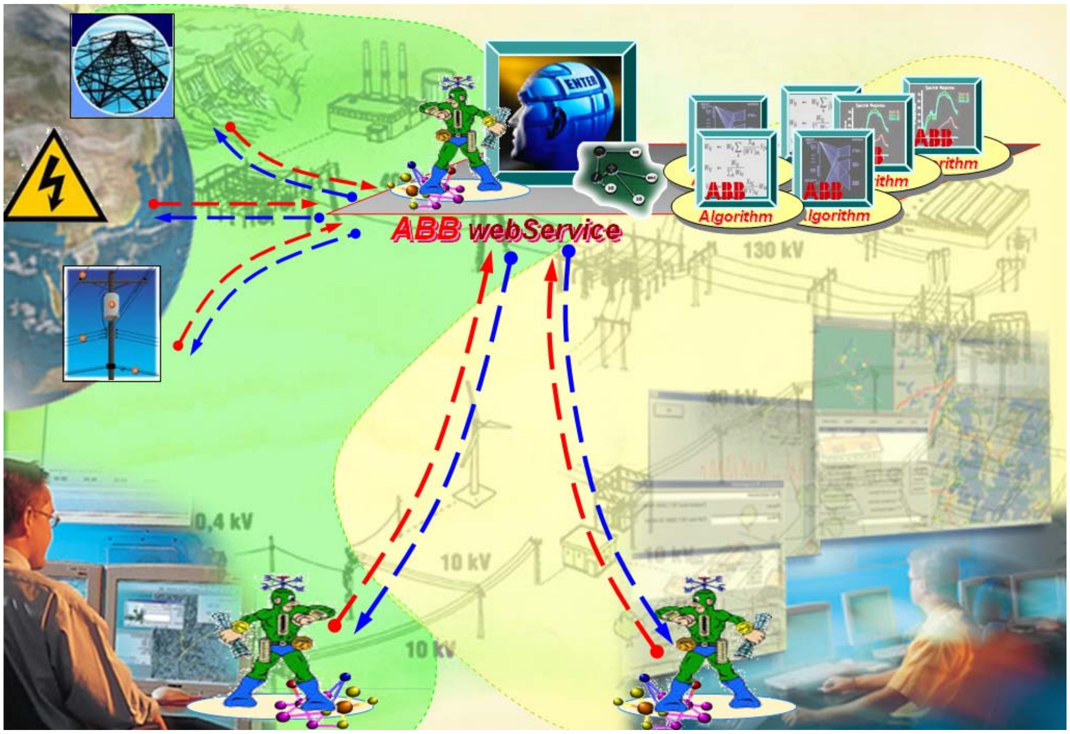


Figure 55 - Scenario: a new business model

GUN platform will ensure interoperability and coordination between such Web-service and customers' software systems, and also a relative ease of implementation of such a solution – because it will not require changes in existing software systems, only extension with GUN. Noticeable that, if semantically defined, such Web-service can potentially be utilized across the globe even by the customers who never purchased any of ABB hardware or software.

The third scenario in our vision is related to the possibility integrating data, which is currently utilized in the power network management (network structure and configuration, feeder relay readings), with contextual information from the external sources (Figure 56). Such integration can be used for:

- *Risk analysis*. Information about whether conditions, ongoing forest works, or forest fires can be used for evaluating existing threats for the power network. This may be used to trigger an alert state for the maintenance team, or even to do a precautionary reconfiguration of the network to minimize possible damage.
- *Facilitation of fault localization*. The output of fault localization algorithms is not always certain. The information about threats for the power network that existed at the time when the fault occurred (which thus may have caused the fault) may greatly facilitate the localization. In some situations, contextual information alone may even be sufficient for localization.
- *Operator interface enhancement.* Contextual information may be used also to extend the operators' view of the power network. For example, satellite imagery can be used for geographic view (instead of locally stored bitmaps as it is in the DMS); also, dynamically-changing information can be accessed and represented on the interface.

Figure 56 - Scenario: integration with external information services

The last scenario is our vision is about the possibility of transferring the knowledge of human experts to automated systems, by means of various data mining tools (Figure 57). In the power network management case, one scenario that seems to be highly appropriate for such knowledge transfer is the following. In present, it is always a decision of a human expert which of the existing fault localization algorithms will perform the best in the context of the existing configuration of the power network and the nature of the fault. Such decisions made by an expert, along with the input data, could be forwarded to a learning Web-service. After a sufficient learning sample, this Web-service could start to be used in some situations instead of the human expert, e.g. in situations when a faster decision is needed or when the expert is unavailable.

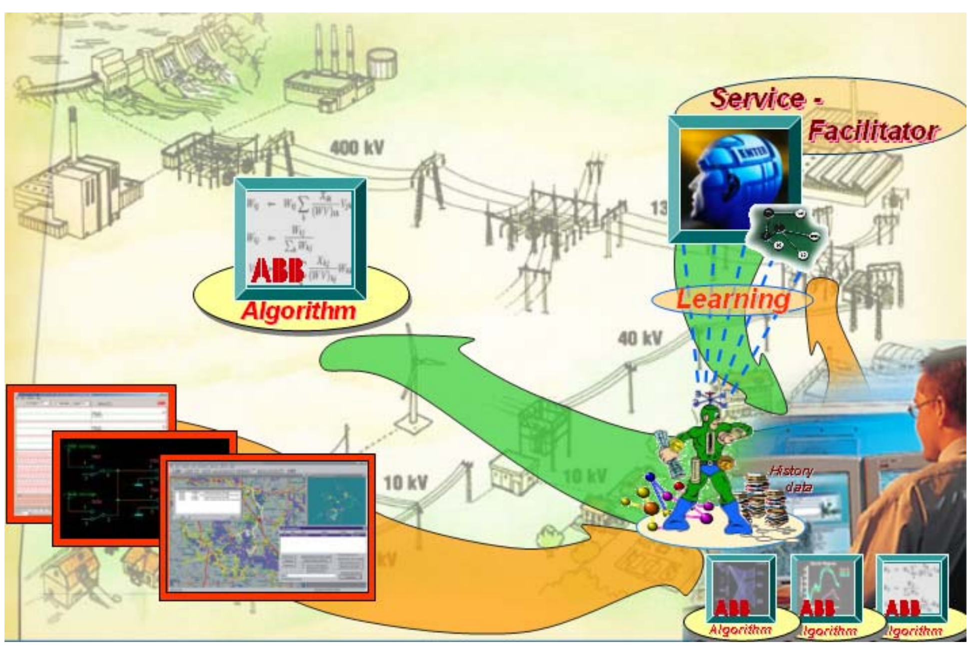


Figure 57 - Scenario: expert's knowledge transfer

9.4.2. The Prototype. We also developed a prototype, mainly for the purpose of the concept demonstration, both for ABB and their customers. The prototype includes the following smart resources, represented by the corresponding agents:

- *Operator*. A human operator monitoring and controlling the power network. In addition to the traditional interfaces – DMS/MicroSCADA – the operator is provided with an additional interface by the operator' agent (see below).
- *Feeders*. Each feeder (section of the power network) is represented by an agent. Those agents are responsible for answering operator's requests for the state of the feeder, and also for sending alerts when a disturbance is registered. Technically, feeder agents are accessing feeders' data from the MicroSCADA system.
- *Network Structure Storage*. The DMS system is utilizing a database for storing the data on the power network including the network graph structure, detailed data on substations, feeders, etc. The network storage agent is responsible for interaction with that database for answering operator's requests for the network graph (for visualization) and e.g. for detailed data on a substation.
- *Fault Localization Services*. The fault localization can be then performed by an external entity, e.g. a Web-service, which will also be represented on GUN platform by a corresponding agent (the service itself is a stub in the prototype).
- *Weather Service*. A service providing constantly updated weather conditions and forecast for a geographic location. We utilized one provided by the Finnish Meteorological Institute.
- *Forest Fire Alert Service*. A service that is supposed to issue alerts when there is a forest fire (a stub in the prototype). The agent representing this service is responsible for automatic forwarding such alerts to the operator's agent.
- *Geographic Service*. Provides the geographic map data in Geography Markup Language (GML), if operator's agent requests.
- *Repository of Roles* and *Pool of Reusable Atomic Behaviors* (RABs).

In the prototype, both the repository of roles and the pool of atomic behaviors are managed by the same agent with the role "OntologyAgent". Also, there is only one single repository of the roles, which is also, in fact, a simplification. Consider the scenario of the fault localization by an external service. The agent representing such a service has to necessarily play at least two different roles. One is "our localization service seller" for the company developed the service, say, ABB. The other is "localization service agent" for the company running a power network, say, Jyväskylän Energia. It is because the agent needs to represent the interest of ABB, sell the service for them; but it is also obliged to deliver the service according to the rules, protocol, etc. specified by Jyväskylän Energia. Obviously, it is reasonable that each of cooperating organizations will maintain its own repository of the roles it defines. However, for a prototype, implementing this was not so important.

Figure 58 shows the process of starting up an agent. The same process is followed for every new agent on the GUN platform. From the startup batch file of the GUN platform, an agent receives only the names of the roles that it has to play (the same holds also for cases when an agent in created in run time). For the example in Figure 58, the agent called "feeder1" gets to know that it has play the general role "FeederAgent" – common for all the feeder agents, and a particular role "FeederID1" – needed for that other agents will associate this agent with the feeder ID1, and including a set of beliefs and rules specific for getting connected to and managing that particular feeder. First, the agent "feeder1" loads the startup.rdf script, which is again common for all the agents. According to that script, the agent contacts the Directory Facilitator to find the agent who plays the "OntologyAgent" role. The Directory

Facilitator maintains a mapping between agents and roles they play. After the OntologyAgent named “ontology” is discovered, it is contacted and asked to deliver the two scripts, one per role needed. After the delivery, “feeder1” loads the scripts and starts to work according to them. It also registers itself with the Directory Facilitator, so that other agents will be aware that it now plays those roles.

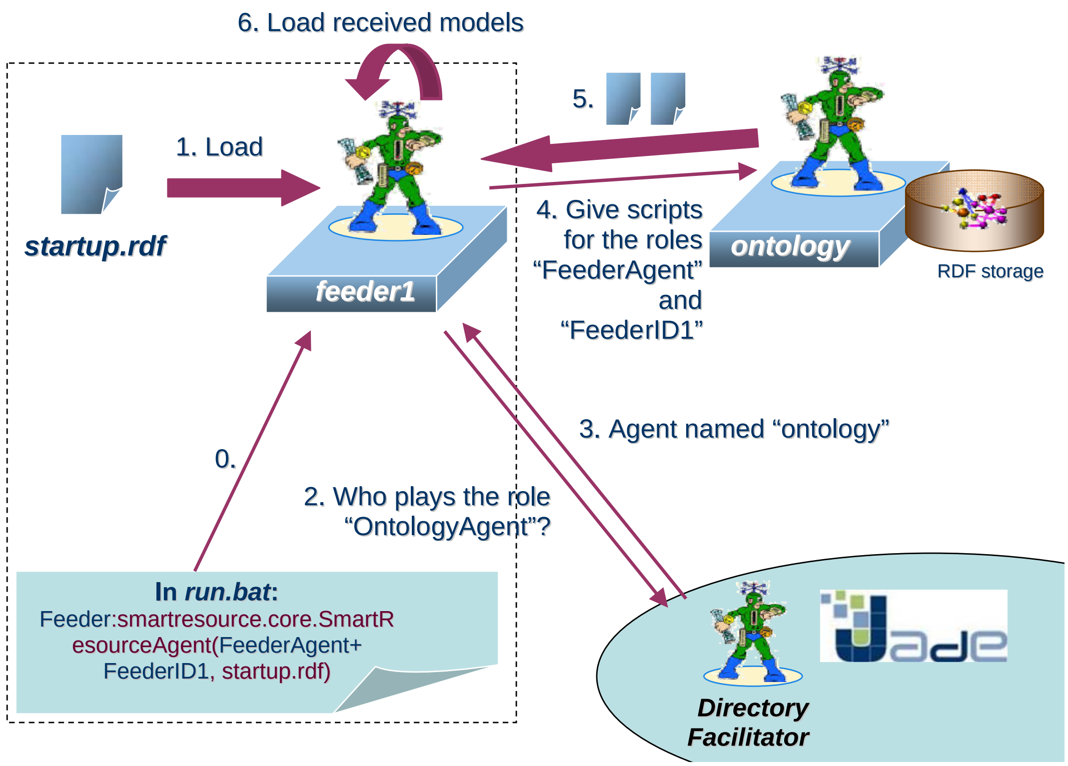


Figure 58 - An agent’s start-up

Figure 59 depicts a more complex scenario of auction for selection of a service provider, in this case a fault localization service. Using the Directory Facilitator, the operator’s agent discovers that there are two competing agents on the platform that provide the fault localization service. The operator’s agent checks its script for a rule resolving such a situation and discovers that, in case of several localization services, an auction has to be performed (for other roles, random select is done). The agent first sends to both localization agents a special request “Load Role OneStepAuctionSeller”, and then a request to make an offer on, say, price of the service. The agent “ls1” has loaded the role “OneStepAuctionSeller” from the beginning, but the agent “ls2” did not. So, “ls2” contacts the OntologyAgent and requests the needed script now. A simple check of rights is performed just before that: with the Directory Facilitator “ls2” checks whether the requesting agent “operator” is working in the role that empowers it to make this particular request, “OperatorAgent” in this case. The agent “ls1” makes its offer immediately, while “ls2” does that after it gets the script and, likely, the corresponding RAB. Then, the operator’s agent selects one of the providers and commits the service transaction with it. This scenario demonstrates that roles can be loaded also dynamically.

Obviously, “ls1” and “ls2” needed to enact the “LocalizationService” role earlier. The behavior model corresponding to it will enable the agent to actually deliver the service (step 12). Also, “ls1” and “ls2” needed to enact some roles like “our service seller” of the corresponding service provider organization. The behavior models of those roles are the places from which they, e.g., get such information as what price to ask from the clients.

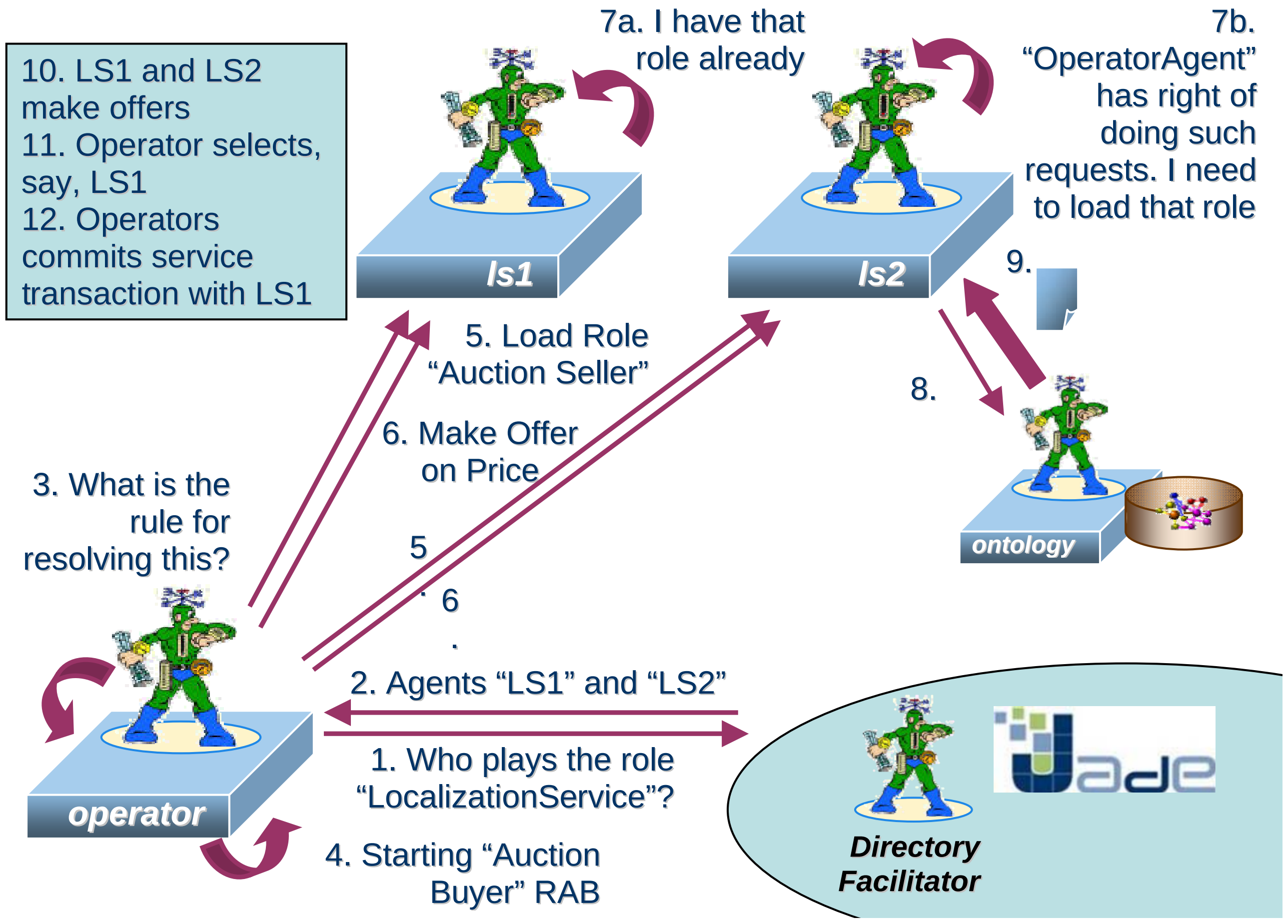


Figure 59 - Auction for selection of the service provider

Figure 60 shows the interface of an operator generated by the operator's agent. The interface consists of the following elements. First, there is a small command window with buttons "Show network in GML", "Show network in GoogleEarth", "Request localization service" and "Send maintenance crew". Second, there is the main graphic interface, which comes in two options. One option utilizes a freeware GML viewer. The other option utilizes the GoogleEarth application, which uses Google's own KML language for defining data to be overlaid over the map. Both GML and KML are XML-based markups, so transition is easy. In the case of using GoogleEarth, participation of the Geographic Service agent is, obviously, not required. The advantage of using GML map data, though, is that it can be used as input for some analysis if needed. For example, one could wish to estimate how the forest fire can progress with time – the information about where lay the boundaries of forests and open spaces or lakes is then important, and may be encoded in GML. In contrast, a satellite image will provide little help in that case.

Finally, the interface may include some other external applications that the operator's agent can pop-up when needed. So, using the main graphic interface, the operator can request the real-time data on the state of a feeder, and the data delivered by the corresponding feeder agent is visualized using the ABB Disturbance Draw application. The operator can also request detailed description of a substation, which will be represented with HTML in an Internet browser window.

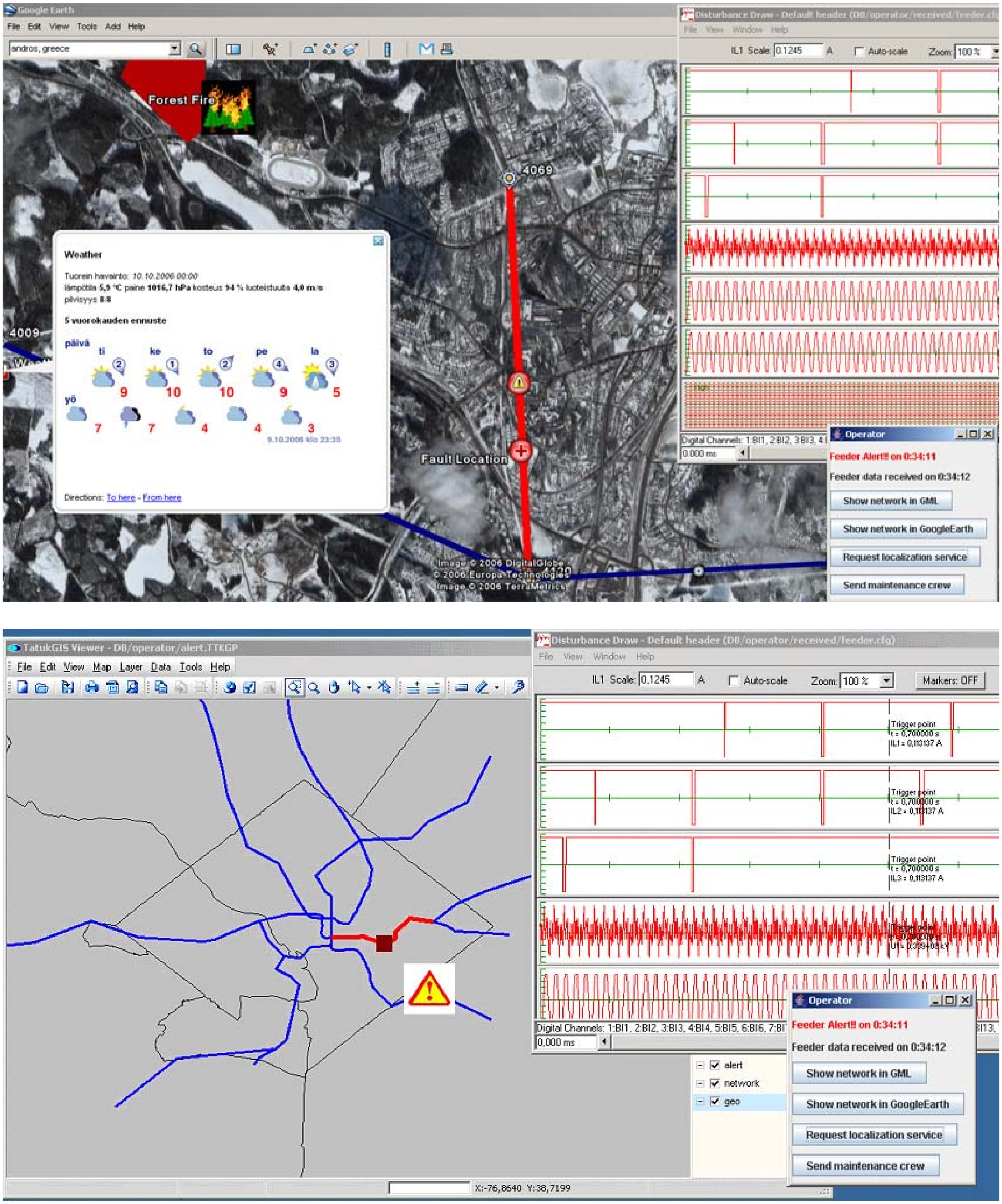


Figure 60 - Interface of an operator provided by his/her agent (2 versions)

10. SmartResource Future

As a continuation of SmartResource, according to our research group roadmap (Figure 1), we are starting activities towards new UBIWARE project. This project supposed to bring the Semantic Web, Distributed AI and Human-Centric Computing to the Ubiquitous Computing domain. It aims at a new generation middleware platform (UBIWARE) which will allow creation of *self-managed* complex industrial systems consisting of mobile, distributed, heterogeneous, shared and reusable components of *different* nature, e.g. smart machines and devices, sensors, actuators, RFIDs, web-services, software, humans, etc. The middleware will enable various components to automatically discover each other and to configure a system with complex functionality based on the atomic functionalities of the components. The project will focus to the following main directions:

1. Design of the core platform that will integrate the semantic description and the agent-based management of UBIWARE resources.
2. Tools for managing and integrating distributed histories of UBIWARE resources.
3. Management of security and privacy in dynamic distributed systems such as UBIWARE.
4. Methods for self-management, configuration and integration of UBIWARE resources.
5. Smart, context-aware GUIs to deliver the integrated data to human UBIWARE users.
6. Middleware for P2P discovery of UBIWARE resources.

The above tasks will be approached by combining various research methods with agile software development processes. This means that software prototypes will be iteratively developed during the whole project lifecycle based on real data, real needs and changing requirements of industrial partners. The result will be both the basic software tools for the UBIWARE platform and several industrial cases prototyped based on these tools.

The UBIWARE project is intended to continue our work towards GUN. The SmartResource project analyzed the central GUN concepts and resulted in some, more or less separated, pilot tools and solutions. In contrast, the UBIWARE project will result in a complete and self-sufficient middleware platform. For this, UBIWARE will integrate SmartResource ideas, elaborate them, and extend with related solutions in supporting but mandatory areas such as security, human interfaces and other.

In this project, we will naturally integrate the Ubiquitous Computing domain with such domains as Semantic Web, Proactive Computing, Autonomous Computing, Human-Centric Computing, Distributed AI, Service-Oriented Architecture, Security and Privacy, and Enterprise Application Integration. We will finish with a real prototype of the UBIWARE for industrial needs as a key toolset for future "Global Enterprise Resource Integration" (GERI) Platform. UBIWARE should bring the following features to industrial partners: Openness, Intelligence, Dynamics, Self-Organization, Seamless Services and Interconnectivity, Flexibility and Reconfigurability, Context-Awareness, Semantics, Proactivity, Interoperability, Adaptation and Personalization, Integration, Automation, Security, Privacy and Trust.

In one sense, our intention to apply the concepts of automatic discovery, selection, composition, orchestration, integration, invocation, execution monitoring, coordination, communication, negotiation, context awareness, etc (which were, so far, mostly related only to the Semantic Web-Services domain) to a more general “Semantic Web of Things” domain. Also we want to expand this list by adding automatic self-management including (self-*)organization, diagnostics, forecasting, control, configuration, adaptation, tuning, maintenance, and learning.

Other closely related plan is to move towards GERI (Figure 1). Modern enterprises face a constantly growing need for interoperability and collaboration of highly *distributed* and heterogeneous ICT systems, both on the intra-enterprise and inter-enterprise levels. Moreover, recent advances in ubiquitous computing allow connecting various physical resources (machines, infrastructure elements, materials, products) to the ICT systems. On one hand, such development intensifies the demand for integration with external resources, such as data storages, information services, and algorithms, which can be found in other units of the same organization, in other organizations, or even on the Internet. On the other hand, it indicates that the traditional concept of Enterprise Application Integration (EAI) must now be extended towards *Global Enterprise Resource Integration (GERI)* that aims at effective and seamless integration of all different types of resources found in an enterprise: digital, physical, and even humans. Two of the major roadblocks for GERI are *heterogeneity* and *complexity*. Heterogeneity hinders the interoperability of the components and makes integration an expensive task. The growing complexity of systems leads to that humans become unable to manage, and sometimes even to anticipate and design, the complexity of interactions, thus calling for *self-manageability* of systems and a high level of (re-)configuration flexibility.

An example scenario: Consider the electrical power distribution domain. Let there be a company (e.g. ABB) providing sensors, products and services for distribution automation worldwide. Let another big company provide enterprise resource management software worldwide (e.g. SAP). There are also quite many companies, which are consumers of these tools, products and software (e.g. Electrical Corporation of Israel), that also use some additional proprietary systems. Can one expect that these products, services and software will be interoperable without significant investments into integration? In our view, GERI should link together not only enterprise business processes (power distribution automation, condition monitoring, predictive maintenance, etc), corporate knowledge (expertise in faults localization, isolation, prediction, diagnostics, risk analysis, annotated monitoring and maintenance data, etc.) and software systems (like ABB's MicroSCADA, SAP, etc.), but also various resources including sensors, power lines, substations, various other devices, and also humans, within and beyond the enterprise. Moreover, appropriate GERI-middleware (e.g. Ubiware) should guarantee two important and challenging requirements: (a) various systems, created with GERI support, should be self-managed and self-configurable (self-configurability here means that configuration of the system can be proactively changed not only by replacing components and services from the set available within an enterprise but also by outsourcing components and services from an external environment); (b) two or more systems created with GERI independently from each other should be interoperable without significant additional efforts.

The future steps are to develop solutions addressing the enterprise interoperability needs based on solid and rigorous scientific theories and principles. The current state of the art puts Distributed AI (i.e. intelligent agents), semantic (meta-) networks, and ontology-based knowledge representation as the main candidates. Most importantly, we believe that the problems of heterogeneity and complexity *cannot* be treated separately. Semantic technologies will facilitate *not only* the discovery of heterogeneous components and data integration across multiple domains, but also the behavioral control and coordination of those components. In our vision, each component to be integrated is represented by an intelligent agent, and the semantics and ontologies are the basis for governing those agents and for the definition and execution of relevant workflows and business processes.

The main objectives of GERI research are:

- To develop architecture for seamless integration of information resources (e.g. software applications, web services) with sensor and RFID devices based on the agents technology leading to a more homogeneous environment.
- To develop tools and methodology for development of distributed systems through integration and reuse of existing heterogeneous components (software or devices), through declarative integration, where the components and their interaction are defined and configured declaratively (semantically) rather than programmatically.
- To elaborate business models of inter-enterprise collaboration through GERI-based environments.
- To analyze application of the GERI approach in Distribution Automation business integration scenarios.

Expected Impact of GERI will be a fully integrated enterprise: GERI will enable *integration globally*, i.e. across the whole enterprise and beyond, and for all types of resources, it will also enable *global integration*, i.e. resources can not only communicate, but also flexibly coordinate with each other, use each other, and jointly engage in different business processes. Reduced costs in integration with legacy systems or development of new systems if some of the relevant components are available. Flexibility is provided in (re-)configuration, agile yet sustainable operation.

11. Related Work

Recent advances in networking, sensor and RFID technologies allow connecting various physical world objects to the IT infrastructure, which could, ultimately, enable realization of the "Internet of Things" and the Ubiquitous Computing visions. Also, this opens new horizons for industrial automation, i.e. automated monitoring, control, maintenance planning, etc, of industrial resources and processes. A much larger, than in present, number of resources (machines, infrastructure elements, materials, products) can get connected to the IT systems, thus be automatically monitored and potentially controlled. Such development will also necessarily create demand for a much wider integration with various external resources, such as data storages, information services, and algorithms, which can be found in other units of the same organization, in other organizations, or on the Internet.

Such interconnectivity of computing and physical systems could, however, become the "nightmare of ubiquitous computing" (Kephart and Chess, 2003) in which human operators will be unable to *manage* the complexity of interactions, neither even architects will be able to *anticipate* and *design* that complexity. It is widely acknowledged that as the networks, systems and services of modern IT and communication infrastructures become increasingly complex, traditional solutions to manage and control them seem to have reached their limits. The IBM vision of autonomic computing (e.g. Kephart and Chess, 2003) proclaims the need for computing systems capable of "running themselves" with minimal human management which would be mainly limited to definition of some higher-level policies rather than direct administration. The computing systems will therefore be *self-managed*, which, according to the IBM vision, includes self-configuration, self-optimization, self-protection, and self-healing.

The vision of autonomic computing emphasizes that the *run-time* self-manageability of a complex system requires its components to be to a certain degree autonomous themselves. Following this, we envision that the software agent technologies will play an important part in building such complex systems. Agent-based approach to software engineering is also considered to be facilitating the *design* of complex systems.

A major problem is inherent *heterogeneity* in ubiquitous computing systems, with respect to the nature of components, standards, data formats, protocols, etc, which creates significant obstacles for interoperability among the components of such systems. Semantic Web technologies are viewed today as a key technology to resolve the problems of interoperability and integration within heterogeneous world of ubiquitously interconnected objects and systems. The Internet of Things should become in fact the *Semantic Web of Things*[24]. Our vision for this project subscribes to this view. Moreover, we believe that Semantic Web technologies can facilitate not only the discovery of heterogeneous components and data integration, but also the behavioral control and coordination of those components.

Self-management of systems is one of the central themes in the EU 7-th Framework ICT Programme (2007-2013). The Objective "Service and Software Architectures" of the Challenge 1 "Network and Service Infrastructures" includes the need for strategies and technologies enabling mastery of complexity, dependability and behavioral stability, and also the need for integrated solutions supporting the networked enterprise. Also, the Objective "The network of the future" of this Challenge includes the need for re-configurability, self-organization and self-management for optimized control, management and flexibility of the future network infrastructure. In addition, the whole Challenge 2 "Cognition, Interaction, Robotics" has as its motivation the need for creating "artificial systems that can achieve general goals in a largely unsupervised way, and persevere under adverse or uncertain conditions; adapt, within reasonable constraints, to changing service and performance requirements, without the need for external re-programming, re-configuring, or re-adjusting". It is noticeable that the systems (stand-alone or networked) monitoring and controlling material or informational processes is one of the three focus areas of this Challenge.

It seems to be generally recognized that achieving the interoperability by imposing some rigid standards and making everyone comply could not be a case in open ubiquitous environments. Therefore, the interoperability requires existence of some *middleware* to act as the glue joining heterogeneous components together. There are a couple of ongoing EU research projects, started in the 6-th FP, that have as one of their goals the development of some middleware for embedded systems. They are RUNES (Reconfigurable Ubiquitous Networked Embedded Systems, 2004-2007)[25] and SOCRADES (Service-Oriented Cross-Layer Infrastructure for Distributed Smart Embedded Devices, 2006-2009)[26]. We believe, however, that the middleware needs of the ubiquitous computing domain go well beyond interconnectivity of embedded systems themselves. There is a more general need for something we refer to as Global Enterprise Resource Integration (GERI), where all different types of resources get seamlessly integrated: physical devices with embedded electronics, web services, software applications, humans along with their interfaces, and other. In the concept of GERI, we also present the need for *true global interoperability, not just interconnectivity*. The components of ubiquitous computing systems should be able not only to communicate and exchange data, but also to flexibly coordinate with each other, discover and use each other, and jointly engage in different business processes.

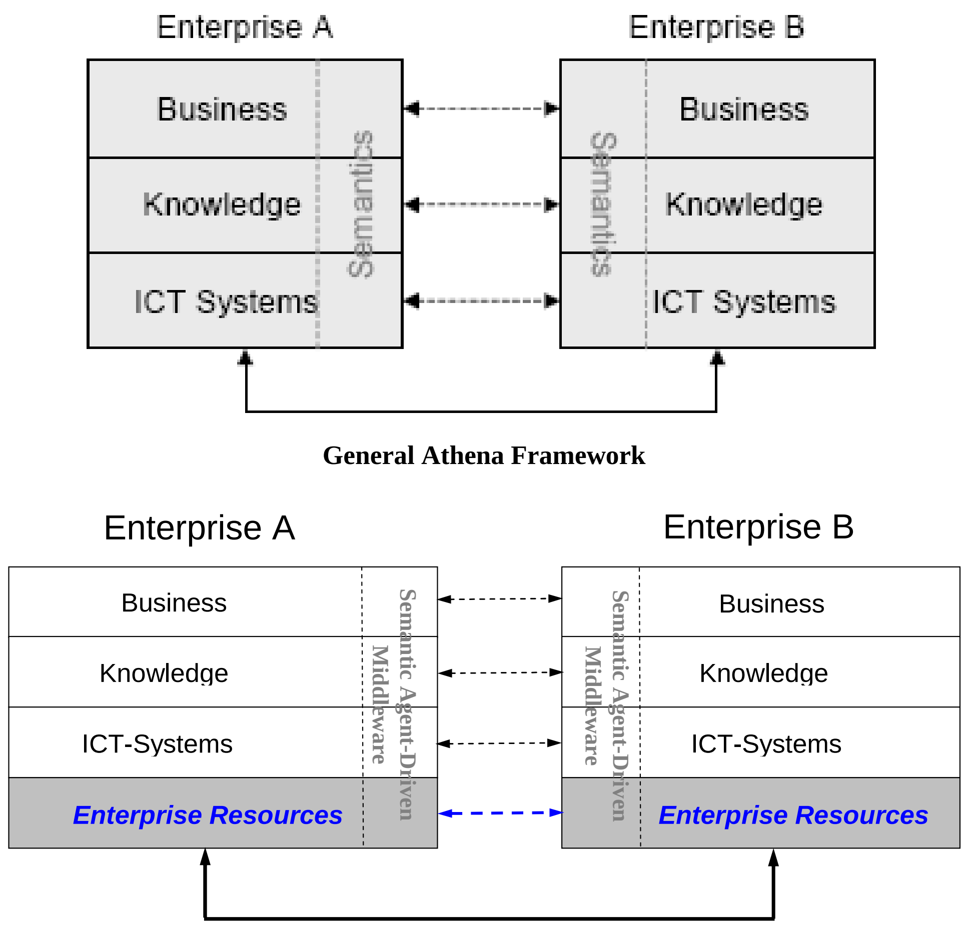
Such more general middleware needs of the ubiquitous computing domain are emphasized in the Strategic Research Agenda (SRA) of the ARTEMIS European Technology Platform[27]. ARTEMIS' SRA has "Seamless Connectivity and Middleware" as one of its three parts. One of the research priorities listed is the middleware as the key enabler for *declarative* programming paradigm, where the

---

[24] David Brock and Ed Schuster (MIT Data Center) at *Semantic Days 2006*, Norway, April 26, 2006, http://www.olf.no/english/news/?30357
[25] RUNES project: http://www.ist-runes.org/
[26] SOCRADES project: http://www.socrades.eu/
[27] ARTEMIS – Advanced Research and Technology for Embedded Intelligence and Systems: http://www.artemis-office.org

components and their interaction are defined and configured declaratively rather than programmatically (and we believe that the semantic technologies are a natural candidate here). The other relevant middleware research priorities include use of *ontologies* for cross-domain systems' organization and for interoperability in heterogeneous environments, dynamic reconfiguration capabilities, adaptive resource management, efficiently bridging information between global, enterprise, and embedded systems, and appropriate security infrastructures.

An FP6 project that is the closest one to GERI in terms of motivation and vision is ATHENA (http://www.athena-ip.org/). See comparison of ATHENA and GERI frameworks in Figure 61. The GERI project will follow and step forward in two main areas. First, ATHENA analyzed enterprise integration on the business, knowledge and ICT systems layers. In a sense, GERI will introduce a fourth layer to the bottom of the ladder, Enterprise Resources layer, and will elaborate on how this layer can be seamlessly integrated with the ICT systems and the higher layers. Second, while ATHENA focused on Service-Oriented Architectures at the ICT systems layer, GERI will elaborate on architectures extending SOA with the agent-based approach. The main reason is that, when getting more physical i.e. considering the Enterprise Resources layer, the issues related to dynamic *coordination* of the system components (e.g. with respect to shared resources) become very important while not treated much in SOA (which is concerned more with the world of web-services or more or less isolated software applications).



General Athena Framework

General GERI Framework

Figure 61 – Comparison of ATHENA and GERI frameworks

An excellent analysis of the current status and the roadmap for the future development of the Internet of Things has been made as collective effort of academy and industry during recent conference organized by DG Information Society and Media, Networks and Communication Technologies Directorate in Brussels (Buckley, 2006). It was pointed out that the Internet of Things characterizes the way that information and communication technologies will develop over the next decade or so. The Internet of Things represents a fusion of the physical and digital worlds. It creates a map of the real world within the digital world. The computer's view of the physical world may, depending on the characteristics of sensor network, possess a high temporal and spatial resolution. The devices on the Internet of Things will have several degrees of sophistication and the final one makes the Proactive Computing (INTEL terminology) possible. These devices (sometimes called Smart Devices) are aware of their context in the physical world and able to react to it, what may cause the context to change.

According to Buckley (2006), the actual power of the Internet of Things arises from the fact that the devices are *interconnected*. Interoperability requires that client of services know the features offered by service providers beforehand and *semantic modeling* should make it possible for service requestors to understand what the service providers have to offer. This is a key issue for moving towards an *open-world* approach, where new or modified devices and services may appear at any time, and towards device networks capable of dynamically adapting to context changes as may be imposed by application scenarios. This has also implications on requirements for middleware, as these are needed to interface between the devices that may be seen as services, and applications. Devices in the Internet of Things might need to be able to communicate with other devices anywhere in the world. This implies a need for a naming and addressing scheme, and means of search and discovery. The fact that devices may be related to an identity (through naming and addressing) raises in turn a number of privacy and security challenges. A consistent set of middleware, offering application programming interfaces, communications and other services to applications, will simplify the creation of services and applications. We need to move from static programming approaches towards a configurable and dynamic composition capability.

In Lassila and Adler (2003), the ubiquitous computing is presented as an emerging paradigm qualitatively different from current personal computing scenarios by involving dozens and hundreds of devices (sensors, external input and output devices, remotely controlled appliances, etc). A vision was presented for a class of devices, so called "Semantic Gadgets", that will be able to combine functions of several portable devices users have today. Semantic Gadgets will be able to automatically configure themselves in new environments and to combine information and functionality from local and remote sources. Semantic Gadgets should be capable of semantic discovery and device coalition formation: the goal should be to accomplish discovery and configuration of new devices without "a human in the loop." Authors pointed out that a critical to the success of this idea is the existence or emergence of certain infrastructures, such as the World Wide Web as a ubiquitous source of information and services and the Semantic Web as a more machine- and automation-friendly form of the Web.

Later, Lassila (2005a, 2005b) discussed possible application of Semantic Web technologies to mobile and ubiquitous computing arguing that ubiquitous computing represents the ultimate "interoperability nightmare". This application is motivated by the need for better automation of user's tasks by improving the interoperability between systems, applications, and information. Ultimately, one of the most important components of the realization of the Semantic Web is "serendipitous interoperability", the ability of software systems to discover and utilize services they have not seen before, and that were not considered when and where the systems were designed. To realize this, qualitatively stronger means of representing service semantics are required, enabling fully automated discovery and invocation, and complete removal of unnecessary interaction with human users. Avoiding a priori commitments about how devices are to interact with one another will improve interoperability and will thus make dynamic, unchoreographed ubiquitous computing scenarios more realistic. Semantic Web technologies are qualitatively stronger approach to interoperability than contemporary standards-based approaches.

To be truly pervasive, the devices in a ubiquitous computing environment have to be able to form a coalition without human intervention. In Qasem et al. (2004), it is noticed that ordinary AI planning for coalition formation will be difficult because a planning agent cannot make a closed-world assumption in such environments. Agent never knows when e.g. it has gathered all relevant information or when additional searches may be redundant. A local closed world reasoning has been incorporated in Qasem et al. (2004) to compose Semantic Web services and to control the search process. The approach has two main components. The first is Plan Generator, which generates a plan that represents a service composition. The second component, the Semantic Web mediator, provides an interface to the information sources, which are devices in the ubiquitous computing environments.

The advances around the Semantic Web and Semantic Web services allow machines to help people to get fully automated anytime, anywhere assistance. However, most of the available applications and services depend on synchronous communication links between consumers and providers. In Krummenacher and Strang (2005), a combination of space-based computing and Semantic Web named as *semantic spaces* is introduced to provide a communication paradigm for ubiquitous services. The semantic spaces approach introduces a new communication platform that provides persistent and asynchronous dissemination of machine-understandable information, especially suitable for distributed services. Semantic spaces provide emerging Semantic Web services and Semantic Gadgets with asynchronous and anonymous communication means. Distributing the space among various devices allows anytime, anywhere access to a virtual information space even in highly dynamic and weakly connected systems. To handle all the semantic data emerging in such systems, data stores will have to deal with millions of triples. In consequence reasoning and processing the data becomes highly time and resource consuming. The solution is to distribute the storage and computation among the involved devices. Every interaction partner provides parts of the space infrastructure and data.

One question is whether Semantic Web is ready to provide services, which fit the requirements of the future Internet of Things? The original idea of Semantic Web (Berners-Lee *et al.*, 2001) is to make Web content suitable not only for human browsing but also for automated processing, integration, and reuse across heterogeneous applications. The effort of the Semantic Web community to apply its semantic techniques in open, distributed and heterogeneous Web environments have paid off: the Semantic Web is evolving towards a real Semantic Web (Sabou *et al.*, 2006). Not only the number of developed ontologies is dramatically increasing, but also the way that ontologies are published and used has changed. We see a shift away from first generation Semantic Web applications, towards a new generation of applications, designed to exploit the large amounts of heterogeneous semantic markup, which are increasingly becoming available. In Motta and Sabou (2006), a number of criteria are given, which Semantic Web applications have to satisfy on their move away from conventional semantic systems towards a new generation of Semantic Web applications:

- *Semantic data generation vs. reuse* (the ability to operate with the semantic data that already exist, i.e. to exploit available semantic markup);
- *Single-ontology vs. multi-ontology systems* (the ability to operate with huge amounts of heterogeneous data, which could be defined in terms of many different ontologies and may need to be combined to answer specific queries);

- *Openness with respect to semantic resources* (the ability to make use of additional, heterogeneous semantic data, at the request of their user);
- *Scale as important as data quality* (the ability to explore, integrate, reason and exploit large amounts of heterogeneous semantic data, generated from a variety of distributed Web sources);
- *Openness with respect to Web (non-semantic) resources* (the ability to take into account the high degree of change of the conventional Web and provide data acquisition facilities for the extraction of data from arbitrary Web sources);
- *Compliance with the Web 2.0 paradigm* (the ability to enable *Collective Intelligence* based on massively distributed information publishing and annotation initiatives by providing mechanisms for users to add and annotate data, allowing distributed semantic annotations and deeper integration of ontologies;
- *Open to services* (the ability of applications to integrate Web-service technology in applications architecture).

In a nutshell, next generation Semantic Web systems will necessarily have to deal with the increased heterogeneity of semantic sources (Motta and Sabou, 2006), which partly corresponds to the trends related to the Internet of Things roadmap for the future development (Buckley, 2006).

One of the most relevant and strongest collaborative efforts (Nokia RC and MIT, Cambridge, Massachusetts) in this area is the recently-started SwapMe[28] (Semantic Web Application Platform for Mobile Ecosystems) project. SwapMe aims at developing a new software architecture that will enable the mobile ecosystem to take maximum advantage of the power of the Semantic Web data model. In order to develop a platform that will enable devices, applications, and agents on the platform to be both context-aware (able to adapt to aspects of the users physical and logical environment) and policy-aware (able to adapt according to semantically-encoded social rules in order to interact with other services, agents and users), SwapMe addresses three major research challenges. The first is applying the Semantic Web data model to the range of devices, services, policies, and data to be available through the Mobile Ecosystem. The second is developing a software architecture that provides a declarative execution model linking the range of resources from data to service to policy through common computational abstractions. The third is eliminating the traditional rigidity of application boundaries that lock data into inflexible presentation models and putting in its place dynamically assembled user interface components that are composed based on user context and policy constraints.

As discussed above, ubiquitous computing systems need explicit semantics for automatic discovery and interoperability among heterogeneous devices. Moreover, it seems that that the traditional Web as such is not enough to motivate the need for the explicit semantics, and this may be a major reason why no "killer application" for the Semantic Web has been found yet. In other words, it is not only that the ubiquitous computing needs Semantic Web, but also the Semantic Web may need the emergence of really ubiquitous computing to finally find its "killer application". Recently, the US Directorate for Computer and Information Science and Engineering (CISE) and National Science Foundation (NSF) has announced an initiative called Global Environment for Networking Innovations (GENI)[29] to explore new networking capabilities and move towards the *Future Internet*. Some of GENI challenges are: support for pervasive computing, bridging physical and cyberspace with the impact to access the information about physical world in real time, and enabling exciting new services and applications (Freeman, 2006). If the Future Internet will allow more natural integration of sensor networks with the rest of the Internet, as GENI envisions, the amount and heterogeneity of resources in the Web will

---

[28] http://research.nokia.com/research/projects/swapme/index.html
[29] GENI: http://www.nsf.gov/cise/geni/

grow dramatically and without their ontological classification and (semi- or fully-automated) semantic annotation processes the automatic discovery will be impossible.

When it comes to developing complex, distributed software-based systems, the *agent-based approach* was advocated to be a well suited one (Jennings, 2001). From the implementation point of view, agents are a next step in the evolution of software engineering approaches and programming languages, the step following the trend towards increasing degrees of localization and encapsulation in the basic building blocks of the programming models (Jennings, 2000). After the *structures*, e.g., in C (localizing data), and *objects*, e.g., in C++ and Java (localizing, in addition, code, i.e. an entity's behavior), agents follow by localizing their *purpose*, the thread of control and action selection. An agent is commonly defined as an encapsulated computer system situated in some environment and capable of flexible, autonomous action in that environment in order to meet its design objectives (Wooldridge, 1997).

However, the actual benefit of the agent-oriented approach arises from the fact that the notion of an agent is also appropriate as a basis for the analysis of the problem to be solved by the system developed. Many processes in the world can be conceptualized using an agent metaphor; the result of such a conceptualization is either a single agent (or cognitive) description or a multi-agent (or social) description (Bosse and Treur, 2006). Jennings (2001) argued that agent-oriented decompositions (according to the purpose of elements) are an effective way of partititioning the problem space of a complex system, that the key abstractions of the agent-oriented mindset are a natural means of modeling complex systems, and that the agent-oriented philosophy for modeling and managing organizational relationships is appropriate for dealing with the dependencies and interactions that exist in complex systems.

The problem of "crossing the boundary" from the domain (problem) world to the machine (solution) world is widely recognized as a major issue in software and systems engineering. Therefore, when it comes to designing software, the most powerful abstractions are those that minimize the semantic distance between the units of analysis that are intuitively used to conceptualize the problem and the constructs present in the solution paradigm (Jennings, 2000). A possibility to have the same concept, i.e. agent, as the central one in both the problem analysis and the solution design and implementation can make it much easier to design a good solution and to handle complexity. In contrast, e.g. the object-oriented approach has its conceptual basis determined by the underlying machine architecture, i.e. it is founded on implementation-level ontological primitives such as object, method, invocation, etc. Given that the early stages of software development are necessarily based on intentional concepts such as stakeholders, goals, plans, etc, there is an unavoidable gap that needs to be bridged. Bresciani et al. (2004) even claimed that the agent-oriented programming paradigm is *the only* programming paradigm that can gracefully and seamlessly integrate the intentional models of early development phases with implementation and run-time phases. In a sense, agent-oriented approach postpones the transition from the domain concepts to the machine concepts until the stage of the design and implementation of individual agents (given that those are still to be implemented in an object-oriented programming language).

Although the flexibility of agent interactions has many advantages when it comes to *engineering* complex systems, the downside is that it leads to *unpredictability in the run time system*; as agents are autonomous, the patterns and the effects of their interactions are uncertain (Jennings, 2000). This raises a need for effective coordination, cooperation, and negotiation mechanism. (Those are in principle distinct, but the word "coordination" is often used as a general one encompassing all three; so for the sake of brevity we will use it like that too.) Jennings (2000) discussed that it is common in specific systems and applications to circumvent these difficulties, i.e. to reduce the system's unpredictability, by using interaction protocols whose properties can be formally analyzed, by adopting rigid and preset organizational structures, and/or by limiting the nature and the scope of the agent interplay. However, Jennings asserted that these restrictions also limit the power of the agent-based approach; thus, in order to realize its full potential some longer term solutions are required.

The available literature sketches two major directions of search for such a longer term solution:

- D1: *Social level* characterization of agent-based systems. E.g. Jennings (2000) stressed the need for a better understanding of the impact of sociality and organizational context on an individual's behavior and of the symbiotic link between the behavior of the individual agents and that of the overall system.
- D2: *Ontological* approaches to coordination. E.g. Tamma et al. (2005) asserted a need for common vocabulary for coordination, with a precise semantics, to enable agents to communicate their intentions with respect to future activities and resource utilization and get them to reason about coordination at run time. Also Jennings et al. (1998) put as an issue to resolve the question about how to enable individual agents to represent and reason about the actions, plans, and knowledge of other agents to coordinate with them.

Recently, some progress has been made with respect to D1, resulting, e.g., in elaboration of the concept of a *role* that an agent can play in an organization (see below). However, with respect to D2 very little has been done. Bosse and Treur (2006) discussed that the agent perspective entails a distinction between the following different types of ontologies: an ontology for internal mental properties of the agent A, MentOnt(A), for properties of the agent's (physical) body, BodyOnt(A), for properties of the (sensory or communication) input, InOnt(A), for properties of the (action or communication) output, OutOnt(A), of the agent, and for properties of the external world, ExtOnt(A). Using this distinction, we could describe the present situation as following. The work on explicitly described ontologies was almost exclusively concerned with ExtOnt(A), i.e. the *domain ontologies*. MentOnt(A) comes for free when adopting a certain agent's internal architecture, such as Beliefs-Desires-Intentions (BDI) (Rao and Georgeff, 1991). Also, the communication parts of InOnt(A) and OutOnt(A) come for free when adopting a certain communication language, such as FIPA's [30] ACL. However, BodyOnt(A), i.e. the perceptors and actuators the agent has, sensory part of InOnt(A), i.e. the agent's perception patterns, and action part of OutOnt(A), e.g. the agent's acting patterns, are not usually treated. However, sharing these ontologies is a necessary precondition for agents' awareness of each other's actions, i.e. for D2. Already referred to article by Tamma et al. (2005) is one of the first endeavors into this direction, which however only introduced and analyzed some of the relevant concepts, such as resource, activity, etc.

In our work, we attempted to provide a solution advancing into both D1 and D2 and somewhat integrating both. Some basic thinking, leading our work, follows.

On the landscape of research in agent-based systems, we can identify two somewhat independent streams of research, each with its own limitations. The first stream is the research in multi-agent systems (MAS); the second stream is the research in agents' internal architectures and approaches to implementation.

Researchers in MAS have contributed with, among others, various methodologies for designing MAS, such as Gaia (Wooldridge et al., 2000), TROPOS (Bresciani et al., 2004), and OMNI (Vázquez-Salceda et al., 2005). For example, OMNI (which seems to be the most advanced with respect to D1)

[30] The Foundation of Intelligent Physical Agents (FIPA), a Standards Committee of the IEEE, http://www.fipa.org/

elaborates on the organizational context of a MAS, defines the relationship between organizational roles and agents enacting those roles, discusses how organizational norms, values and rules are supposed to govern the organization's behavior and thus to put restrictions on individual agents' behaviors. However, OMNI touches only on a very abstract level the question about how the individual agents will be implemented or even function; the agents are treated as rather atoms. One reason is that it is (reasonably) assumed that the agent organization's designer may have no direct control over the design of individual agents. The organization designer develops the rules to be followed and enforcing policies and entities, such as "police" agents, while development of other agents is done by external people or companies. One of few concrete implementation requirements mentioned in OMNI is that a rule interpreter must be created that any agent entering the organization will incorporate, somehow. The OMNI framework also includes explicitly the ontological dimension, which is restricted, however, to a domain ontology only (see the previous section), and thus does not provide much new with respect to D2.

The other stream of research, on individual agents, has contributed e.g. with well-known BDI architecture, and several *agent programming languages* (APL) such as AgentSpeak(L) (Rao, 1996), 3APL (Dastani et al., 2003) and ALPHA from the AgentFactory framework (Ross and Collier, 2004). All of those are declarative languages and based on the first-order logic of n-ary predicates. For example, an agent program in ALPHA consists of declarations of the beliefs and goals of that agent and declaration of a set of rules, including belief rules (generating new beliefs based on current ones), reactive rules (invoking some actions immediately) and commitment rules (adopting a commitment to invoke an action). Perceptors (perceiving environment and generating new beliefs) and actuators (implementing the actions to be invoked) are then pieces of external code, in Java. As discussed in the previous section, agent-oriented approach postpones the transition from the domain concepts to the machine concepts until the stage of the design and implementation of individual agents. The advantage of using an APL like ALPHA is that the transition is postponed even further, until the implementation of particular perceptors and actuators.

This advantage seems to be, however, the only one that is considered. We did not encounter in literature approaches that would extend the role of APL code beyond the development stage. APL code is assumed to be written by the developer of an agent and either compiled into an executable program or interpreted in run-time but remaining an agent's intrinsic and static property. APL code is not assumed to ever come from outside of the agent in run-time, neither shared with other agents in any way.

Such export and sharing of APL code would, however, probably make sense in the light of findings from the field of MAS, and also in the light of D2. Methodologies like OMNI describe an organizational role with a set of rules, and an APL is a rule-based language. So, using an APL for specifying a role sounds as a natural way to proceed. The difference is that APL code corresponding to a role should naturally be a property of and controlled by the organization, and accessed by the agents' enacting the role potentially even in the run-time. Run-time access would also enable the organization to update the role code if needed.

The second natural idea is that the agents may access a role's APL code not only in order to enact that role, but also in order to coordinate with the agents playing that role. As one option, an agent can send to another agent a part of its APL code to communicate its intentions with respect to future activities (so there is no need for a separate content language). As another option, if a role's code is made public inside the organization, the agents may access it in order to understand how to interact with, or what to expect from, an agent playing that role.

However, when thinking about using the existing APLs in such a manner, there are at least two issues:

- The code in an APL is, roughly speaking, a text. However in complex systems, a description of a role may need to include a huge number of rules and also a great number of beliefs representing the knowledge needed for playing the role. Also, in a case of access of the code by agents that are not going to enact this role, it is likely that they may wish to receive only a relevant part of it, not the whole thing. Therefore, a more efficient, e.g. a database-centric, solution is probably required.
- When APL code is provided by an organization to an agent, or shared between agents, mutual understanding of the meaning of the code is obviously required. While using first-order logic as the basis for an APL assures understanding of the semantics of the rules, the meaning of predicates used in those rules still needs to be consistently understood by all the parties involved. On the other hand, we are unaware of tools allowing unambiguous description of the precise semantics of n-ary predicates.

As a solution to these two issues, we see using an APL-like language based on the Resource Description Framework (RDF), which is a central technology of Semantic Web. RDF uses binary predicates only, i.e. triples (n-ary predicates can be represented nevertheless, of course, using several approaches). For RDF, tools are available for efficient database storage and querying, and also for explicit description of semantics, e.g. using OWL. Such an RDF-based APL-like language was developed by our group earlier (Kaykova *et al.*, 2005c). We refer to it *RgbDF* meaning "resource goal and behavior description framework". In the near future, however, one of the important tasks is developing it further.

12. Conclusions

In summary, current scientific trends, emerging US initiatives, and concerns of the EU ICT strategy indicate that the research challenges related to interoperability of resources in ubiquitous environments are becoming more and more important. The current state of the art in ICT puts Semantic Web, Distributed AI (i.e. agents) and Human-Centric Computing as main candidates for acting as core technologies for building middleware providing such interoperability. Our group has significant expertise in these technologies, and, most importantly, in using them jointly. This is our main motivation for putting forward our research roadmap: SmartResource-UBIWARE-GERI-GUN. We believe that if the goal is to guarantee the European competitiveness relatively to fast developing US initiatives and also to further develop Finnish innovations in the field of ubiquitous computing, such a roadmap could be considered as *just in time*. Our group has relevant experience, the results from the SmartResource (2004-2006)[31] project, and thus can be seen as having some time advantage relatively to others in this area.

The SmartResource project had results having certain business potential. A convincing evidence of this is that Metso Automation has started a privately-financed project with our group to further advance the prototype developed for them during the project. The SmartResource project analyzed the central concepts related to our vision, and resulted in some, more or less separated, pilot tools and solutions. In contrast, the incoming UBIWARE and GERI project initiatives will result in a complete and self-

[31] SmartResource project Web site: http://www.cs.jyu.fi/ai/OntoGroup/SmartResource_details.htm

sufficient middleware platform. We are positive, therefore, that UBIWARE and GERI will have even more significant business potential.

Acknowledgements

We are grateful to Tekes (National Technology Agency of Finland), Agora Center of the University of Jyväskylä, and to cooperating and contributing companies (ABB, Metso Automation, TeliaSonera, TietoEnator, and Jyväskylä Science Park) for supporting activities of the SmartResource project. Special thanks to industrial partners and their contact persons for collaboration on the case studies reported in this paper, and permission to publish the results. We are also grateful for Agora Center (University of Jyväskylä) and its personnel for being an excellent host for our project. Thanks to Prof. Timo Tiihonen for support and practical help provided to Industrial Ontologies Group team during project period and beyond.

REFERENCES

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

Bosse, T., and Treur, J. (2006) Formal interpretation of a multi-agent society as a single agent. *Journal of Artificial Societies and Social Simulation* 9(2).

Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., and Mylopoulos, J. (2004) Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems* 8(3): 203-236

Buckley, J. (2006) From RFID to the Internet of Things: Pervasive Networked Systems, Final Report on the Conference organized by DG Information Society and Media, Networks and Communication Technologies Directorate, CCAB, Brussels.
Available in: http://www.rfidconsultation.eu/docs/ficheiros/WS_1_Final_report_27_Mar.pdf .

Dastani, M., van Riemsdijk, B., Dignum, F., and Meyer, J.-J.Ch. (2003) A programming language for cognitive agents: Goal directed 3APL. *Proc. First International Workshop on Programming Multi-Agent Systems, LNCS vol. 3067*, pp. 111-130.

Freeman, P. A. (2006) Statement before the Committee on Science of the U.S. House of Representatives, Hearing on Innovation and Information Technology: The Government, University and Industry Roles in Information Technology Research and Commercialization, Austin, Texas (available in: http://www.house.gov/science/hearings/full06/May%205/Freeman.pdf).

Jennings, N.R., Sycara K. P., and Wooldridge, M. (1998) A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems* 1(1): 7-38.

Jennings, N.R. (2000) On agent-based software engineering. *Artificial Intelligence* 117(2): 277-296.

Jennings, N.R. (2001) An agent-based approach for building complex software systems. *Communications of the ACM* 44(4): 35-41.

Kaykova *et al.* (2005a). Kaykova, O., Khriyenko, O., Kovtun, D., Naumenko, A., Terziyan, V. and Zharko, A., General Adaption Framework: Enabling Interoperability for Industrial Web Resources, International Journal on Semantic Web and Information Systems, Idea Group, 1(3), 31-63.

Kaykova *et al.* (2005b). Kaykova, O., Khriyenko, O., Naumenko, A., Terziyan, V. & Zharko, A., RSCDF: A Dynamic and Context-Sensitive Metadata Description Framework for Industrial Resources, Eastern-European Journal of Enterprise Technologies, 3(2), 55-78.

Kaykova *et al.* (2005c). Kaykova, O., Khriyenko, O., Terziyan, V. & Zharko, A., RGBDF: Resource Goal and Behaviour Description Framework, In: M. Bramer and V. Terziyan (Eds.): Industrial Applications of Semantic Web, Proceedings of the 1-st International IFIP/WG12.5 Working Conference IASW-2005 , August 25-27, 2005, Jyvaskyla, Finland, Springer, IFIP, 83-99.

Kaykova *et al.* (2006). Kaykova O., Khriyenko O., Terziyan V., General Networking Framework, *Technical Report (Deliverable D 3.1)*, SmartResource Tekes Project, Agora Center, University of Jyvaskyla, January-May, 2006. Available in: http://www.cs.jyu.fi/ai/OntoGroup/Restricted/D3_1.pdf

Khriyenko, O. & Terziyan, V. (2004). 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, 175-179.

Khriyenko O., and Terziyan V. (2006), A Framework for Context-Sensitive Metadata Description, In: *International Journal of Metadata, Semantics and Ontologies,* Inderscience Publishers**,** ISSN 1744-2621, 2006, Vol. 1, No. 2, pp. 154-164.

Kephart J. O. and Chess D. M. (2003) The vision of autonomic computing, *IEEE Computer*, Vol. 36, No. 1, pp. 41-50.

Krummenacher, R., and Strang, T. (2005) Ubiquitous Semantic Spaces, In: Conference Supplement to the 7th Int'l Conf on Ubiquitous Computing (UbiComp 2005), Tokyo.

Lassila, O. (2005a) Applying Semantic Web in Mobile and Ubiquitous Computing: Will Policy-Awareness Help?, in Lalana Kagal, Tim Finin, and James Hendler (eds.): Proceedings of the Semantic Web Policy Workshop, 4th International Semantic Web Conference, Galway, Ireland, pp. 6-11.

Lassila, O. (2005b) Using the Semantic Web in Mobile and Ubiquitous Computing, in: Max Bramer and Vagan Terziyan (eds.): Proceedings of the 1st IFIP WG12.5 Working Conference on Industrial Applications of Semantic Web, Springer IFIP, pp. 19-25.

Lassila, O., and Adler, M. (2003) Semantic Gadgets: Ubiquitous Computing Meets the Semantic Web, In: D. Fensel et al. (eds.), Spinning the Semantic Web, MIT Press, pp. 363-376.

Motta, E., and Sabou, M. (2006) Next Generation Semantic Web Applications, In: Proceedings of the 1st Asian Semantic Web Conference (ASWC), Beijing, China.

Naumenko *et al.* (2005). Naumenko A., Nikitin S., Terziyan V., Zharko A., SmartResource Prototype Environment, Version 2.0, *Technical Report (Deliverable D 2.3)*, SmartResource Tekes Project, Agora Center, University of Jyvaskyla, August-December, 2005.
Available in: http://www.cs.jyu.fi/ai/OntoGroup/Restricted/D2_3.pdf .

Nikitin *et al.* (2005). 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.): *Autonomous Intelligent Systems: Agents and Data Mining, Proceedings of the AIS-ADM-05*, June 6-8, 2005, St. Petersburg, Russia, Springer, LNAI 3505, pp. 200-214.

Nikitin *et al.* (2007). Nikitin S., Terziyan V., Pyotsia J., Data Integration Solution for Paper Industry, In: *Proceedings of the 4th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, May 9-12, 2007, Angers, France, Springer, 13 pp. (submitted 9 January 2007).

Qasem, A., Heflin J., and Mucoz-Avila H. (2004) Efficient Source Discovery and Service Composition for Ubiquitous Computing Environments. In: Workshop on Semantic Web Technology for Mobile and Ubiquitous Applications, ISWC 2004.

Rao, A.S. and Georgeff, M.P.(1991) Modeling rational agents within a BDI architecture. Proc. 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91), pp. 473-484.

Rao, A.S. (1996) AgentSpeak(L): BDI agents speak out in a logical computable language. *Proc. 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, LNCS vol.1038*, pp. 42-55.

Ross, R.J. and Collier, R. (2004) *The ALPHA Programming Language: Language Guide*. AgentFactory technical reports. Online http://www.agentfactory.com/publications/ alpha_lang_guide.pdf

Sabou, M., Lopez, V., and Motta, E. (2006) Ontology Selection on the Real Semantic Web: How to Cover the Queens Birthday Dinner?, In Proceedings of EKAW, Podebrady, Czech Republic.

Salmenjoki *et al.* (2007). Salmenjoki K., Tsaruk Y., Terziyan V., Viitala M., Agent-Based Approach for Electricity Distribution Systems, In: *Proceedings of the 9-th International Conference on Enterprise Information Systems*, 12-16, June 2007, Funchal, Madeira, Portugal, 8 pp. (submitted 23 November 2006).

Tamma, V.A.M., Aart, C., Moyaux, T., Paurobally, S., Lithgow-Smith, B., and Wooldridge, M. (2005) An ontological framework for dynamic coordination. *Proc. 4th International Semantic Web Conference'05, LNCS vol. 3729*, pp. 638-652.

Terziyan, V. (2005), Semantic Web Services for Smart Devices Based on Mobile Agents, International Journal of Intelligent Information Technologies, Idea Group, 1(2), 43-55.

Terziyan, V. (2003). 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: OTM 2003 Workshops*, Springer-Verlag, LNCS, 2889, 279-291.

Terziyan, V., & Puuronen, S., (1999). Knowledge Acquisition Based on Semantic Balance of Internal and External Knowledge, *Lecture Notes in Artificial Intelligence*, 1611, 353-361.

Terziyan V., & Katasonov A. (2007). Terziyan V., Katasonov A., Global Understanding Environment: Applying Semantic Web to Industrial Automation, In: J. Cardoso, M. Hepp, M. Lytras (eds.), *Real-world Applications of Semantic Web Technology and Ontologies,* Springer, 2007, 31 pp. (Book chapter, submitted 23 October 2006).

Vázquez-Salceda, J., Dignum, V., and Dignum, F. (2005) Organizing multiagent systems. *Autonomous Agents and Multi-Agent Systems* 11(3): 307-360.

Wooldridge, M. (1997) Agent-based software engineering. *IEEE Proceedings of Software Engineering* 144(1): 26-37.

Wooldridge, M., Jennings, N.R., Kinny, D. (2000) The Gaia Methodology for Agent- Oriented Analysis and Design. *Autonomous Agents and Multi-Agent Systems* 3(3): 285-312.

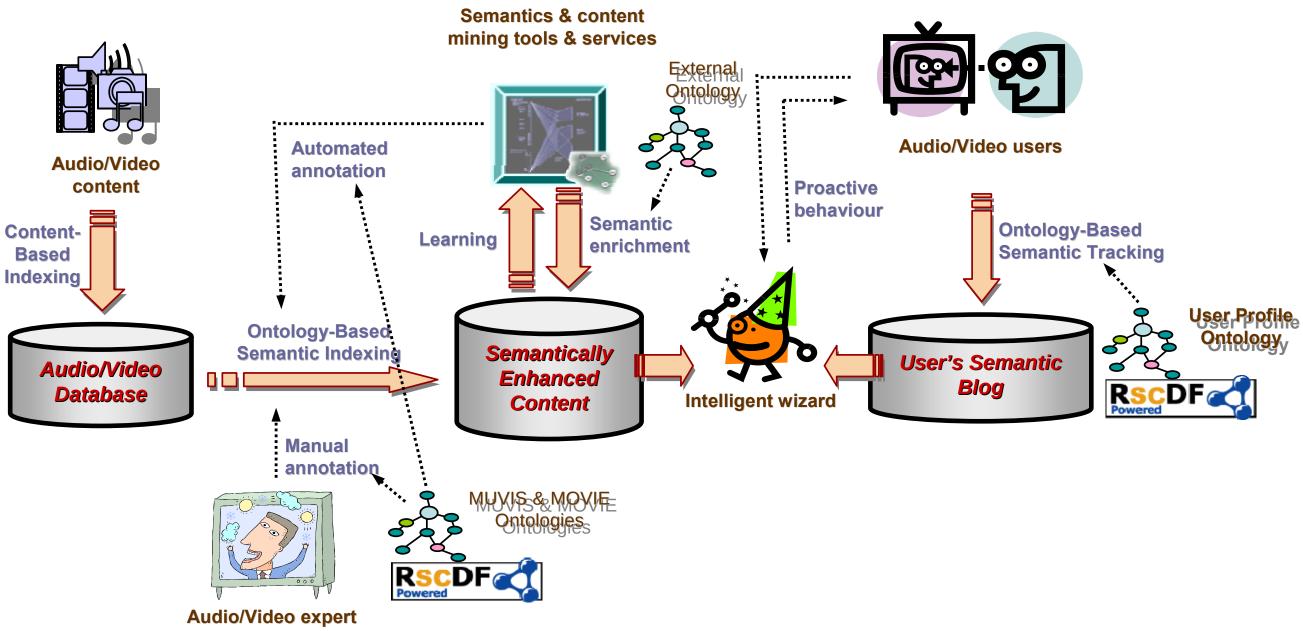
SmartResource

Cooperation & Networking

M-Advantage

M-Advantage*:* "Multimedia – Automatic Digital Video and Audio Network through Advanced Publishing European Service".

Proposal to 6th Framework Program (FP6-027131, 22.03.2005), 15 partners (558 person/months for first 18 months of the project), Finnish Node: ***TUT + IOG.*** Contact person – ***Moncef Gabbouj***.



SODIUM

SODIUM: "Service-Oriented Development in a Unified Framework" (http://www.atc.gr/sodium/) 6th Framework Specific Targeted Project (IST-FP6-04559, started 1 July 2004), 6 partners (4 countries), Liaison with SmartResource via ***Aphrodite Tsalgatidou***.

The objective of SODIUM is to develop a standards-based, comprehensive solution for the integration of heterogeneous services. This project aims to bridge this gap by developing a collection of models, languages and open source corresponding middleware to support the comprehensive visual service composition, analysis, execution, management and monitoring of heterogeneous (web, p2p, and grid) services, in an open and unified manner. The project partners are: ***ATC (GR), NKUA (GR), SINTEF (N), ETH (CH), LOCUS (N) and MEDISYSTEM (R).***

***SODIUM and SmartResource*:** The two projects use similar ideas for handling heterogeneous services and devices. More specifically, SODIUM uses plug-ins to access the various heterogeneous services and registries. In this way, the SODIUM platform becomes independent of the existing heterogeneous registries and services. On the other hand, SmartResource uses generic semantic adapters (General Adaptation Framework) and supporting ontology (Resource State/Condition Description Framework) for providing interoperability between different types of industrial resources: devices, software components (services) and humans (operators or experts). Therefore, an obvious cooperation is to share experiences for the philosophy and the internal architecture of adapters and plug-ins. Additional useful cooperation can be related to analysis and development of registries and registry mediators for eServices and web resources in general. The projects will remain in close touch and there will be a link from each project site to the other.

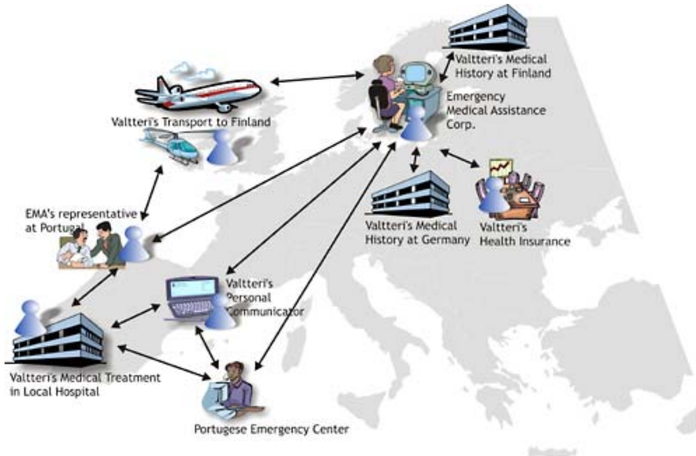
SODIUM-2: "Service-Oriented Development in a Unified Framework" (http://www.atc.gr/sodium/) 6th Framework Specific Targeted Project (submitted September 2005), 6 partners (4 countries), ***IOG*** one of the partners.

CASCOM

CASCOM: "Context-Aware Business Application Service Coordination in Mobile Computing Environment" (http://www.ist-cascom.org/) 6th Framework Specific Targeted Project (FP6 – 511632, started 1 September 2004), 8 partners (7 countries), Liaison with SmartResource via ***Heikki Helin***.

The main objective of the project is to implement value-added supportive infrastructure for business application services for mobile users across mobile and fixed networks. The driving vision of CASCOM is that ubiquitous business application services are flexibly coordinated and pervasively provided by intelligent agents in dynamically changing contexts of open, large-scale, and pervasive environments.

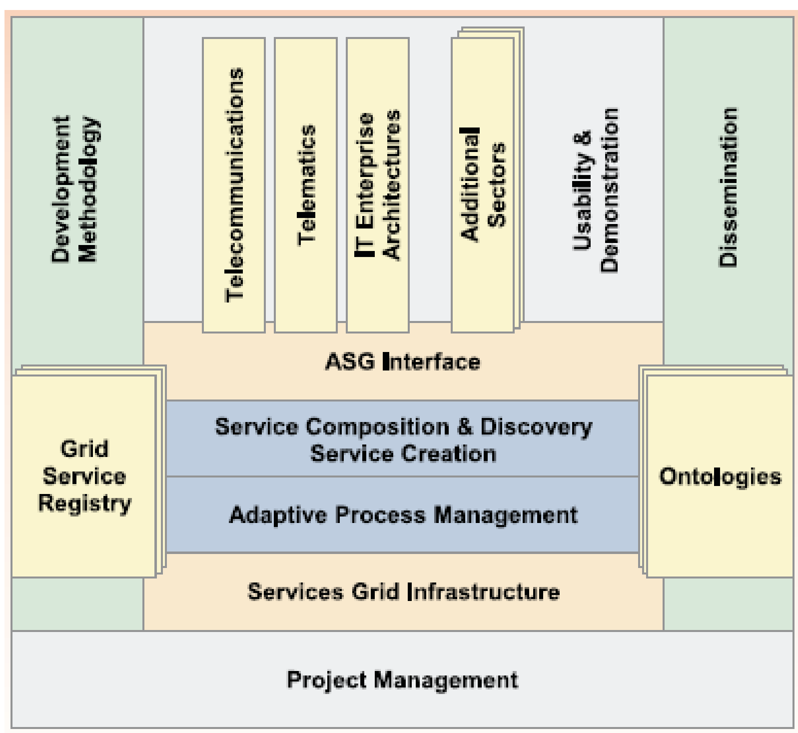
One step toward the realization is the development of an intelligent agent-based peer-to-peer (IP2P) environment. IP2P environments are extensions to conventional P2P architectures with components for mobile and ad hoc computing, wireless communications, and a broad range of pervasive devices. A major challenge in IP2P environments is to guarantee a secure spread of (personal) service requests across multiple transmission infrastructures and ensure the trustworthiness of services that may involve a broad variety of providers.



ASG

ASG: "Adaptive Services Grid" (http://asg-platform.org/cgi-bin/twiki/view/Public) 6th Framework Integrated Project (IST-FP6-04617 started 01.09.2004), 21 partners (7 countries), Finnish Node: ***TITU*** (University of Jyvaskyla) in cooperation with ***IOG***. Contact person – ***Jari Veijalainen***.

The objective of the ASG project is the development of an open platform for adaptive services, discovery, creation, composition, and enactment and the development of an open platform for the European telecommunication industry to faster the rapid roll out of new services and new products. From a business point of view, ASG aims to make processes and services more effective by using semantic information to fulfil user requests, by enabling services provisioning with service components from different platforms and suppliers, by enabling dynamic adaptive service composition and by providing high scalable platform using grid technologies.



Scope of ***SmartResource*** – ***ASG*** cooperation: Ontologically Enhanced Smart Registry Platforms for Grid Services.

IOG contribution: 9 men/months of work for ASG deliverable (***A. Naumenko, S. Nikitin***) and supervising 2 Master Theses of ASG team members.

MODPA

MODPA: "Mobile Design Patterns and Architectures" (http://www.titu.jyu.fi/modpa/index.htm) Tekes Project started 01.05.2004), Place: ***TITU*** (University of Jyvaskyla) in cooperation ***with Nokia, Yomi Software, Tieturi, SESCA Technologies, Metso Paper, Trusteq.*** Contact person – ***Jouni Markkula***.

MODPA is a research project aiming at improving and increasing the efficiency of mobile application and service development by the means of design patterns.

Current and developing mobile technologies provide a rich environment for developing advanced mobile applications and services. However, the multitude of architectural solutions makes the design environment highly complex. Design patterns, earlier used primarily in object oriented programming, are method for introducing and communicating efficient and verified solutions for design problems in complex environment. Design patterns can be a highly useful tool for speeding and improving mobile application and service development. In the MODPA project, the application of design patterns for supporting mobile application and service development is studied.

Scope of ***SmartResource – MODPA*** cooperation: Access Control Based on Ontology of Security Patterns.

IOG contribution: 4 men/months of work for MODPA deliverable (***A. Naumenko***).

KRM Group

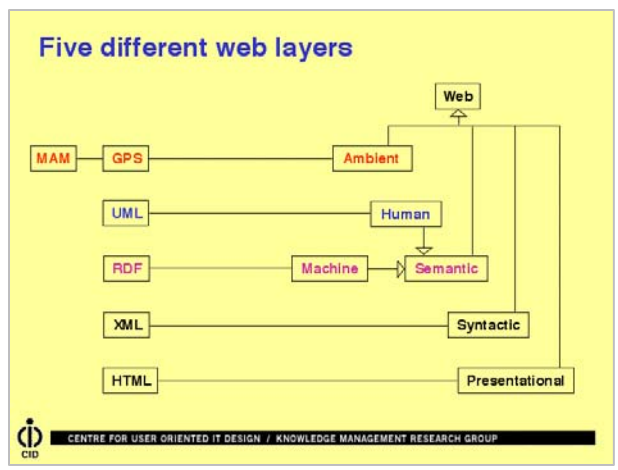
KRM Group: "Knowledge Management Research Group" (http://kmr.nada.kth.se/) Centre for User-Oriented IT Design, partners from Sweden, Norway, Finland, Greece, Spain, Romania, Ukraine, Finnish Node: ***IOG***.

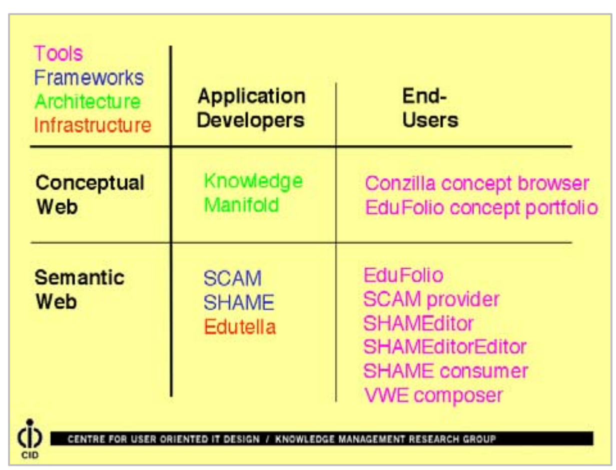
The KMR group is a research group driven by a desire to create new and powerful ways to *structure and communicate information* in order to support its *transformation into knowledge* and *transmutation into understanding*. Problem domains include educational, industrial as well as administrative settings.

Information architecture that called a *Knowledge Manifold* (KM) consists of a number of linked information landscapes (contexts), where one can navigate, search for, annotate and present all kinds of electronically stored information. A KM is constructed by *conceptual modeling* of a specific knowledge domain in order to capture its underlying *thought patterns* in the form of context-maps.

The group supposes to use the KM architecture to construct a *conceptual web* which should function as a human (semantic) interface to the underlying (machine) *semantic web*.

When used in education, the KM-architecture represents a paradigm shift, away from the traditional teacher-centric, curriculum-based "knowledge-pushing" *educational design patterns* towards a more learner-centric, "knowledge-pulling" educational architecture based on learner interest and curiosity.





MODE

MODE: "Management of Distributed Expertise in R&D Collaboration" Regional Development Project, ***Vaasa University + International Partners + IOG***. Contact person – ***Kimmo Salmenjoki***.

The goal of MODE is to create a unified and wide action plan for technology oriented development projects with in the Ostrobothnia region. *Hierarchical models will be established, through which different actors can find each others in the various collaboration networks*. The methodologies will be developed and the best practices for finding and advocating innovations will be utilized. This will be established through more long standing operation models based on the presently existing and used project oriented models and collaborations with the local SME companies and their workers (our adult students) and partners.

The focus will be primarily on web based technologies and their application to the further enhancement of project based work groups and schemes taken back to the organizational (for summary and further planning) level.

Yaroslav Tsaruk from ***IOG*** went to Vaasa to work for MODE for 3 months.

Topics for cooperation: knowledge management and service oriented software usage in industrial information systems; from operability and data integration questions towards more unified approaches in sharing the data and the interoperability of related information systems.

SCOMA

SCOMA: "Scientific Computing and Optimization in Multidisciplinary Applications" (http://www.jyu.fi/agora-center/en/research/scoma.shtml), Tekes Application to MASI Technology Program. ***IOG has own Workpackage: "SCOMA Semantic Web Portal".*** Contact person – ***Pekka Neittaanmaki, Timo Tiihonen.***

The activities focus on adding explicit semantics to the target mathematical resources, which will be integrated to the SCOMA knowledge portal. Semantics will be added in a form of metadata represented according to the standards of W3C's Semantic Activity. Metadata will serve to facilitate the administration of the libraries of mathematical knowledge, the search and retrieval of mathematical knowledge and the reuse of the knowledge by different mathematical applications.

The SCOMA Semantic Web portal has two main purposes. First of all, it will provide a national tool for advanced publishing, sharing and reuse of mathematical tools, expertise and knowledge distributed among the heterogeneous SCOMA parties and their contacts (mainly Finnish industries and research organizations at this stage). The portal will also serve as the first pilot of a more comprehensive European initiative to be undertaken by ECCOMAS (international project starting in 2007).

An approach will involve Web Service standards and Service-oriented analysis combined with Semantic Web approach. This kind of approach would automate the process of composition of complex FEM software for specific tasks through integration of heterogeneous atomic software FEM components published at SCOMA Semantic Web portal.

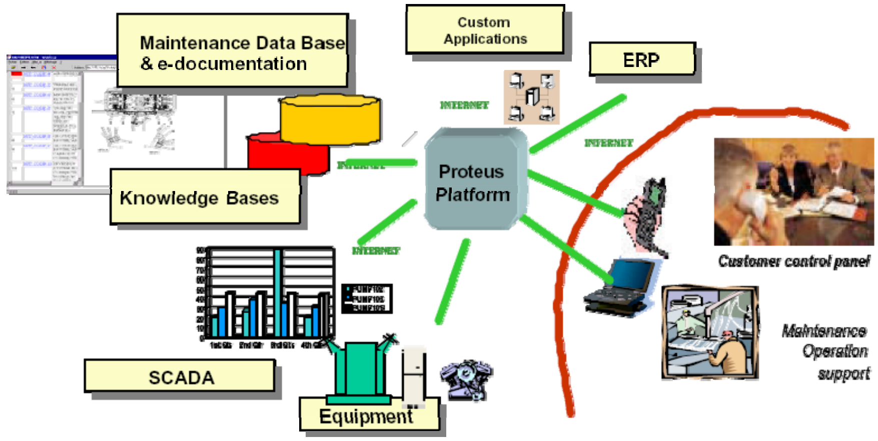
Proteus

Proteus: "A Generic Platform for e-Maintenance" (http://www.proteus-iteaproject.com/), ITEA Project, Coordinating Company – Cegelec.

PROTEUS developing European generic software architecture for web-based e-maintenance centers, targeting the transportation, energy and other industries. The aim is to improve efficiency by bringing expertise via Internet directly to the user site.

PROTEUS promotes a de-facto form of standardization through extensive use of new data-structuring technologies (**XML** - Extensible Mark-up Language), application integration techniques and Internet-related technologies. This will reduce maintenance process costs (time to diagnosis and duration of intervention), and also prevent failures through early monitoring of field equipment (condition-based predictive maintenance).

Computerized Maintenance Management System tools are key in Maintenance, Repair & Operation management. None are genuinely open, although ERP (Enterprise Resource Planning) environments such as SAP provide useful extensions. As a considerable effort is required to make these autonomous tools communicate, there is a clear need for an integrated environment to enable newly developed and existing products to work together. Enabling technologies are needed to build a global platform for SCADA (Supervisory Control And Data Acquisition) systems, condition monitoring, CMMS, ERP and e-procurement tools, and decision-support tools based on rules and CBR (Case-Based Reasoning).

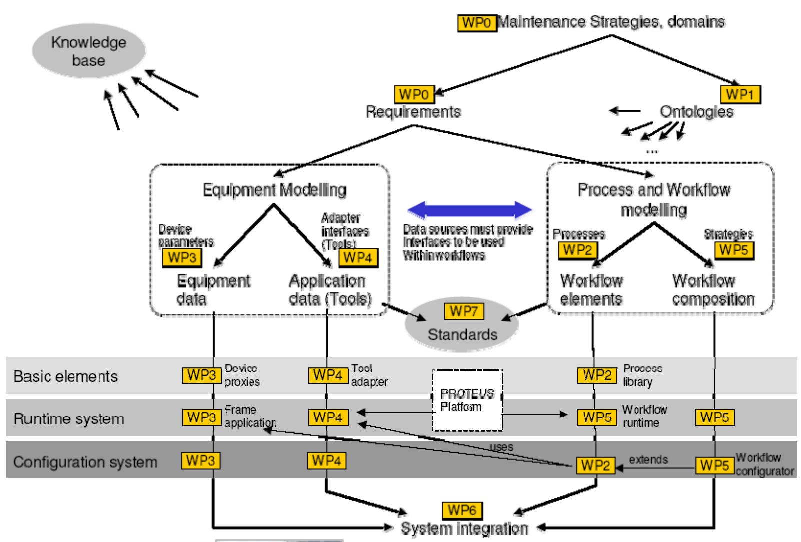


SHIVA

SHIVA: “Automatic Generation of e-Maintenance Platform Dedicated to the Equipment to Maintain” (***continuation of Proteus project***), ITEA Project Proposal, Coordinating Company – Cegelec Cigma, France.

European industrial companies are facing fierce competition in an open economy, and increasing pressure is put on maintenance to keep production equipment at its maximum capacity in terms of quantity, quality of products throughout its lifetime, at a controlled cost. Maintenance is becoming a co-operative task, involving multiple actors from multiple organizations. These actors must handle increasing complexity of production equipment and increasing heterogeneity of data and software tools. Then the problem to maintenance actors is to be able to handle a lot of different tools to optimize and perform their maintenance process. This is often combined with an error-prone manual data transfer between the applications.

The SHIVA initiative aims at enabling fast and easy creation of equipment modeling, so as to generate PROTEUS based e-maintenance platform in any industrial domain. This project will contribute to an improvement of the usability of the plant equipment and efficiency of maintenance activities. This will be reached through the business process driven integration of tools (CMMS, SCADA, ERP), the generation of dedicated maintenance platform from the equipment and process models, an advanced user friendly commissioning procedures based on a systematic equipment and process modeling approach.



Invitation to join SHIVA

Industrial Ontologies Group

Denis Reboul, Cegelec Cigma (France) – PROTEUS and SHIVA project manager.

International Conference *IASW-2005*

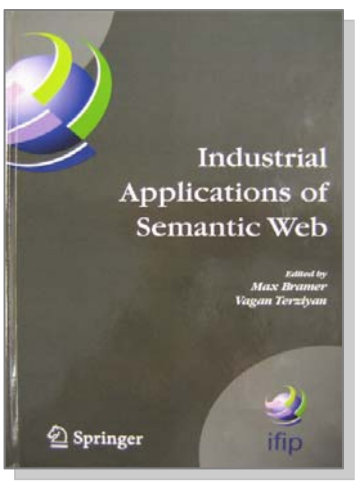
Industrial Applications of Semantic Web *(IASW-2005)*

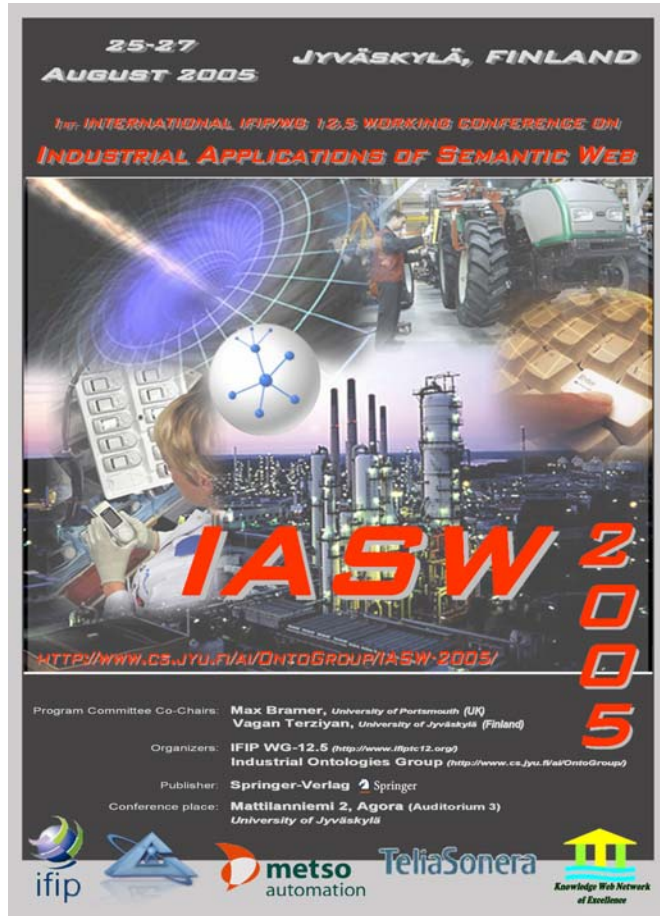
(1st International IFIP/WG 12.5 Working Conference IASW-2005, 25-27 August, Jyväskylä, Finland)

http://www.cs.jyu.fi/ai/OntoGroup/IASW-2005/main.html

IASW-2005: Really International Conference (Contributors and participants from 20 countries worldwide):

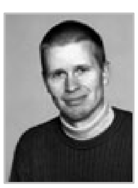
- Australia
- Austria
- China
- Estonia
- Finland
- France
- Germany
- Greece
- Ireland
- Italy
- Japan
- Korea
- Mexico
- Moldova
- Netherlands
- Russia
- Senegal
- Spain
- Ukraine
- USA





Conference is organized by ***Industrial Ontologies Group***

Key-note talks:



Ora Lassila

(Research Fellow & Head of Competence Area, Nokia Research Center, Boston, USA)

"Using the Semantic Web in Mobile and Ubiquitous Computing"

Alain Leger

(Senior Researcher, France Telecom R&D, France)

"Semantic Web Applications: Fields and Business Cases. The Industry Challenges the Research"

Amit P Sheth

(Professor, Large Scale Distributed Information Systems Lab, University of Georgia, USA)

"Enterprise Applications of Semantic Web: the Sweet Spot of Risk and Compliance"

MIT Data Center

We will cooperate on study "***M-Language vs. OWL/RDF***" from various angles:

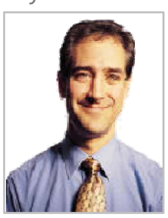
(a) M-Language as complementary to OWL/RDF (where RDF considered as such and also as extended to RscDF, RgbDF, RpiDF);
(b) OWL/RDF as complementary to M-Language;
(c) Semantic adapter M-Language - OWL/RDF;
(d) Semantic adapter OWL/RDF - M-Language;
(e) M-Language + OWL/RDF = towards Global RFID;
(f) M-Language +OWL/RDF+ Mathematical Models + UBIWARE = Semantic Web of Things.

1. Building an **M Adaptor to Semantic Web technologies** such as RDF or OWL. In this application, keyed words from the M dictionary would be used as part of OWL to enable improved merging of graphs. We think this would improve the capabilities of OWL in situations where data or knowledge from different industries is integrated.

2. The **Semantic Web of Things**. Industry is still in the preliminary stages of understanding how to use RFID data to link physical objects together. The M Language has potential as a means of building these links in a semantically unambiguous way. This has great potential to information sharing between companies and to improve other supply chain capabilities such as track and trace.

3. **Network of Mathematical Models**. Enables unique opportunity to automatically discover, utilize and integrate mathematical models (published or automatically created) for complex automation tasks.

4. **North-European MIT Data Center**. Both MIT and IOG have an interest in creating such branch based on IOG premises to manage cooperative activities and utilize MIT experiences in Scandinavia and beyond.



David L. Brock
Founder and Director,
The Data Center
Principal Research Scientist,
MIT
dlb@mit.edu

Edmund W. Schuster
Co-director,
The Data Center
Principal Research Scientist,
MIT
edmund_w@mit.edu

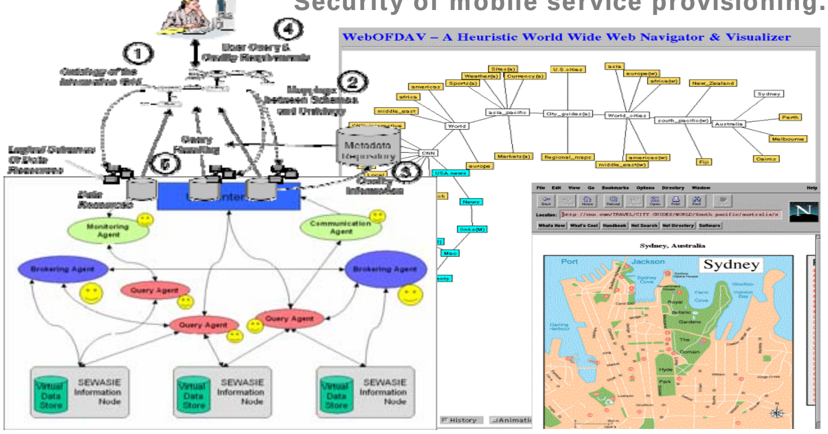
Berkeley

User modeling of activity integrating various sensing (e.g. microphones, GPS, occupancy sensors, etc.) and building applications on top.

Professor John Canny,
Berkeley Institute of Design,
University Berkeley, California

Aachen

Security of mobile service provisioning.



Professor Matthias Jarke,
Information Systems Group,
RWTH Aachen University,
Germany

Cooperation & Networking

Technion, Israel

We will continue cooperation with Prof. Gregory Levitin (Israel Electric Corporation and Technion) on risk management, reliability analysis and optimization in ubiquitous environments. As our project's industrial consortium contains ABB and Fingrid, a significant experience of Electric Corporation of Israel in predicting and avoiding risks in power networks is of great importance to our project. During the meetings of our group representatives with the Electric Corporation of Israel experts in Haifa (November 2006), the company interest for cooperation in intelligent distribution automation domain has been confirmed and the parties currently are preparing plans for concrete actions.

Reliability and risk analysis, system survivability enhancement, reliability engineering, reliability optimization, optimal defense strategy against intentional attacks.

Professor Gregory Levitin,
Faculty of Industrial Engineering & Management,
Technion - Israel Institute of Technology
Engineer-Expert. Reliability & Equipment Department. R&D Division.
The ***Israel Electric Corporation Ltd***

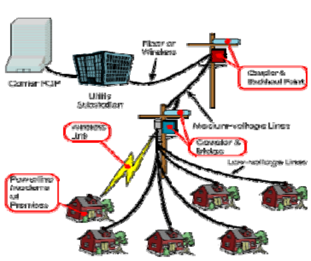
ITIN, France

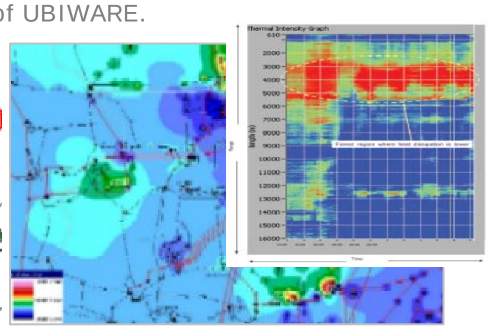
Integrating of Semantic Web and Agent Technologies with Telecommunications and Mobile Computing and its utilization in Business and Education.

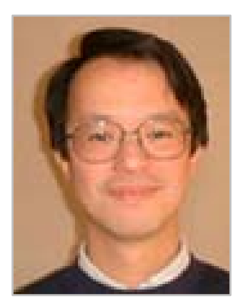
Prof. Alain Gourdin
Managing Director
ITIN, IT-Institute
Cergy-Pontoise cedex, France
www.itin.fr

Okayama University, Japan

We have also agreed on cooperation with Prof. Hiroshi Suito (Solid Waste Management Research Center, Okayama University, Japan) in areas of simulation, self-descriptive visualization and optimization of the content in relation to the Smart Interfaces workpackage of UBIWARE.







Prof. Hiroshi Suito
Solid Waste Management Research Center, Okayama University, Japan
http://www.ems.okayama-u.ac.jp/

Kharkov National University of Radioelectronics, Ukraine

Our ongoing cooperation with Kharkov National University of Radioelectronics (Ukraine) adds to the project a unique expertise in intelligent information processing, machine learning and data mining needed for processing semantic history repositories.

Other cooperation within Europe

Industrial Ontologies Group has contacts with VU Amsterdam (Prof. Jan Treur, Prof. Frank van Harmelen, Dr. Borys Omelayenko) with expertise in multi-agent systems, Semantic Web and ontology learning; DERI Ireland/Austria with expertise in integration and composition of Web and grid services.

Other

KnowledgeWeb: (http://knowledgeweb.semanticweb.org/) EU Network of Excellence (FP6-507482 started 1.01.2004), 22 partners, ***IOG*** applied for partnership and got grant for ***IASW-2005***.

East-East OSI: "Modern Technologies for Developing Teaching Resources for Web Teaching Systems" Open Society Institute, A Soros Foundations Network Grant***, Yerevan University*** (Armenia) + ***Kharkov National University of Radioelectronics*** (Ukraine) + ***IOG***. Contact person – ***Vagan Terziyan***.

SmartUniversity: "Creating Smart Platforms for Management of University Academic and Research Collaborative Activities", Initiative of Finnish Ministry of Education. Contact person and negotiator – ***Timo Tiihonen***.

CITIC-2005: (http://www.ciisca.org/citic05/) "International Congress on Information Technologies", Initiative of Ingenieria de Sistemas, University of Cuenca, & CIISCA, Ecuador. Contact person – ***Andriy Zharko*** and ***Dr. Villie Morocho***.

SmartCook: To develop a new family of applications for mobile phones users to help them to handle their everyday activities and technologically to enable interoperation between the applications based on semantic resource description and domain ontology. Partners: ***IOG*** + ***IOG (Kharkov Filial)***. Contact persons – ***Oleksiy Khriyenko*** and ***Oleksandr Shevchenko***.

SmartWell-Being: "WELL-BEING AND LIFE EXPECTANCY IMPROVEMENT SYSTEM", To create system of granting of the individual services providing improvement of quality of a life, on the basis of use of modern means of telecommunications, advanced information, diagnostic and biotechnologies. Partners: ***IOG*** + ***Telemedicine Research Group*** (Kharkov). Contact persons – ***Volodymyr Kushnaryov*** and ***Vagan Terziyan***.

SmartResource

Publications

! *There are some publications dated by the year 2003. These are related to the project preparation, planning and application writing period and are relevant to the project. The list also includes few publications, which belong to the scope of cooperation between SmartResource project team and other projects or organizations.*

Edited Books and Book Chapters

1. Terziyan V., Katasonov A., Global Understanding Environment: Applying Semantic Web to Industrial Automation, In: J. Cardoso, M. Hepp, M. Lytras (eds.), *Real-world Applications of Semantic Web Technology and Ontologies,* Springer, 36 pp. (Book chapter, submitted 30 October 2006).
2. Terziyan V., Challenges of the "Global Understanding Environment" based on Agent Mobility, In: V. Sugumaran (ed.), *Advanced Topics in Intelligent Information Technologies*, Vol. 1, Idea Group, 2007, pp.121-152. (to appear).
3. V. Ermolayev, N. Keberle, O. Kononenko, V. Terziyan, Proactively Composing Web Services as Tasks by Semantic Web Agents, In: L.J. Zang (Ed.), *Modern Technologies in Web Services Research*, Idea Group Inc., 2006, 31 pp., (book chapter, to appear).
4. Ermolayev V., Terziyan V., Kaykova O., SW @ UKRAINE, In: M. Lytras (Ed.), *Semantic Web Fact Book 2005*, AIS SIGSEMIS, 2006, ISSN: 1556-2301, pp. 13-17.
5. Kaykova O., Khriyenko O., Kovtun D., Naumenko A., Terziyan V., Zharko A., Challenges of General Adaptation Framework for Industrial Semantic Web, In: A. Sheth and M. Lytras (eds.), *Advanced Topics in Semantic Web*, Idea Group, Vol. 1, 33 pp. (Book chapter, to appear).
6. Bramer M., Terziyan V. (eds), Industrial Applications of Semantic Web, *Proceedings of the 1st International IFIP/WG12.5 Working Conference on Industrial Applications of Semantic Web*, Springer IFIP, Vol.188, 2005, ISBN: 0-387-28568-7, 340 pp.

Refereed Journal Papers

7. Naumenko A, Srirama S., Terziyan V., Jarke M., Semantic Authorizations of Mobile Web Services, In: *Computer Communications Journal, Special Issue on Security on Wireless Ad Hoc Networks*, 2006, Elsevier, ISSN: 0140-3664, 28 pp. (submitted 18 December 2006).
8. Terziyan V., Bayesian Reasoning based on Predictive and Contextual Feature Selection, In: *International Journal on Artificial Intelligence Tools*, Special Issue, 2006, World Scientific, ISSN: 0218-2130, 20 pp. (Extended version of SETN-2006 conference paper, submitted 28 September 2006).
9. Pulkkinen, M., Naumenko, A., and Luostarinen, K., Managing Information Security in a Business Network of Machinery Maintenance Services Business - Enterprise Architecture as a Coordination Tool, , In: Sangkyun Kim (Ed.), *Special Issue on Methodology of Security Engineering for Industrial Security Management Systems, Journal of Systems and Software*, ELSEVIER, (In press).
10. Khriyenko O., Terziyan V., A Framework for Context-Sensitive Metadata Description, In: *International Journal of Metadata, Semantics and Ontologies*, ISSN 1744-2621, 2006, Vol. 1, No. 2, pp. 154-164.
11. Veijalainen J., Terziyan V., Tirri H., Transaction Management for M-Commerce at a Mobile Terminal, In: *Electronic Commerce Research and Applications, Special Issue on Mobile Technology and Services*, Vol. 5, No. 3, 2006, Elsevier, ISSN: 1567-4223, pp. 229-245.
12. Naumenko A., Nikitin S., Terziyan V., Service Matching in Agent Systems, In: *M.S. Kwang (Ed.), Special Issue on Agent-Based Grid Computing, International Journal of Applied Intelligence*, Vol. 25, No. 2, 2006, ISSN: 0924-669X, pp. 223-237.
13. Naumenko A., Nikitin S., Terziyan V., Zharko A., Strategic Industrial Alliances in Paper Industry: XML- vs. Ontology-Based Integration Platforms, In: *The Learning Organization, Special Issue on: Semantic and Social Aspects of Learning in Organizations*, Emerald Publishers, ISSN: 0969-6474, 2005, Vol. 12, No. 5, pp. 492-514.

14. Kaykova O., Khriyenko O., Kovtun D., Naumenko A., Terziyan V., Zharko A., General Adaption Framework: Enabling Interoperability for Industrial Web Resources, In: *International Journal on Semantic Web and Information Systems*, Idea Group, ISSN: 1552-6283, 2005, Vol. 1, No. 3, July-September 2005, pp. 31-63.

15. 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.

16. Terziyan V., A Bayesian Metanetwork, In: *International Journal on Artificial Intelligence Tools, World Scientific,* Vol. 14, No. 3, 2005, World Scientific, ISSN: 0218-2130, pp. 371-384.

17. 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. 2, 2005, ISSN: 1729-3774, pp. 55-78.

18. Khriyenko O., "SemaSM: Semantically enhanced Smart Message", In: *Eastern-European Journal of Enterprise Technologies*, Vol. 1, No. 13, 2005, ISSN: 1729-3774.

19. Terziyan V., Zharko A., Semantic Web and Peer-to-Peer: Integration and Interoperability in Industry, In: *International Journal of Computers*, Systems and Signals, ISSN 1608-5655, Vol. 4, No. 2, 2003, pp. 33-46.

20. Terziyan V., Khriyenko O., Mobile Agent-Based Web Service Components in Semantic Web, In: *Eastern-European Journal of Enterprise Technologies*, Vol. 2, No. 2, 2004, ISSN: 1729-3774, pp. 4-15.

21. Levykin V., Terziyan V., Shevchenko A., Organization and Technology for Educating Specialists at a Department based on Knowledge Management, In: *Eastern-European Journal of Enterprise Technologies*, Vol. 2, No. 2, 2004, ISSN: 1729-3774, pp. 15-19 (in Russian).

22. 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 , 2004, pp. 63-87.

23. 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, 2004, ISSN: 1729-3774, pp. 37-49.

Refereed International Conference Papers

24. Salmenjoki K., Tsaruk Y., Terziyan V., Viitala M., Agent-Based Approach for Electricity Distribution Systems, In: *Proceedings of the 9-th International Conference on Enterprise Information Systems*, 12-16, June 2007, Funchal, Madeira, Portugal, 8 pp. (submitted 23 November 2006).

25. Nikitin S., Terziyan V., Pyotsia J., Data Integration Solution for Paper Industry, In: J.P. Müller and K. Mertins (Eds.), In: *Proceedings of the 3rd International Conference on Interoperability for Enterprise Software and Applications (IESA-07)*, March 28-30, 2007, Madeira Island, Portugal, 13 pp. (submitted 2 November 2006).

26. Naumenko A., SEMANTICS-BASED ACCESS CONTROL: Ontologies and Feasibility Study of Policy Enforcement Function, In: J. Filipe and J. Minguillon (Eds.), In: *Proceedings of the 3rd International Conference on Web Information Systems and Technologies (WEBIST-07),* March 3-6, 2007, Barcelona, Spain, 6 pp.

27. Naumenko A., Katasonov A., Terziyan V., A Security Framework for Smart Ubiquitous Industrial Resources, In: J.P. Müller and K. Mertins (Eds.), In: *Proceedings of the 3rd International Conference on Interoperability for Enterprise Software and Applications (IESA-07),* March 28-30, 2007, Madeira Island, Portugal, 13 pp. (to appear).

28. Khriyenko O., Coordination of the Distributed Proactive Smart Resources, In: *Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN 2007) as part of the 25th IASTED International Multi-Conference on APPLIED INFORMATICS*, Innsbruck, Austria, 13-15 February, 2007, 7pp.

29. Naumenko, A. and Luostarinen, K., 2006, Access Control Policies in (Semantic) Service-Oriented Architecture, In Schaffert S. and Sure Y. (Eds.), *Semantic Systems From Visions to Applications, Proceedings of the SEMANTICS 2006*, Austrian Computer Society, Vienna, Austria, pages 49-62.

30. Naumenko, A., Contextual rules-based access control model with trust, In Shoniregan C. A. and Logvynovskiy A. (Eds.), *Proceedings of the International Conference for Internet Technology and Secured Transactions, ICITST 2006*, 11-13 September, London, UK, ISBN 0-9546628-2-2, e-Centre for Infonomics, pages 68-75.

31. Luostarinen, K., Naumenko, A., Pulkkinen, M., 2006, Identity and Access Management for Remote Maintenance Services in Business Networks, in IFIP *International Federation for Information Processing, Volume 226, Project E-Society: Building Bricks*, eds. R. Suomi, Cabral, R., Hampe, J. Felix, Heikkilä, A., Järveläinen, J., Koskivaara, E., (Boston: Springer), pp. 1-12.

32. Terziyan V., Bayesian Metanetwork for Context-Sensitive Feature Relevance, In: G. Antoniou et al. (eds.), *Proceedings of the 4-th Hellenic Conference on Artificial Intelligence (SETN 2006*), Lecture Notes in Artificial Intelligence, Vol. 3955, 2006, pp. 356-366.

33. Tsaruk Y., Salmenjoki K., Vanio A., Uden L., Arumugam G. Semantic and context-aware message sharing between organizations, *KMO-2006 Knowledge management in organization*, 13-14 June 2006, Maribor, Slovenia.

34. Salmenjoki K., Tsaruk Y., Vainio A., Multi-agent approach for knowledge processes between research organizations*, KMO-2006 Knowledge management in organization,* 13-14 June 2006, Maribor, Slovenia.

35. Mazhelis, O. and Naumenko, A., The Place and Role of Security Patterns in Software Development Process, In Medina, E.F. and Yagüe, M.I. (Eds.), *Security in Information Systems Proceedings of the 4th International Workshop on Security in Information Systems, WOSIS 2006 In conjunction with ICEIS 2006* Paphos, Cyprus, May 2006. INSTICC Press, pp. 91-100.

36. Jari Veijalainen, Sergiy Nikitin, Ville Tormala, "Ontology-based Semantic Web Service platform in Mobile Environments," /mdm/, p. 83, *7th International Conference on Mobile Data Management (MDM'06),* 2006.

37. Tsaruk Y., Salmenjoki K., Dispert H., Using agent approach in developing a course sharing system, Information Society 2005: *Intelligent systems*, October 10-17, 2005, Ljubljana, Slovenia.

38. Tsaruk Y., Salmenjoki K., Semantic Web Approaches for Networked Collaboration, In: G. Papadourakis and I. Lazaridis (Eds.): *New Horizons in Industry, Business and Education, Proceedings of the 4th International Conference NHIBE-2005,* August 25-26, 2005, Corfu, Greece, ISBN : 960-85316-9-1 pp. 316-321.

39. Naumenko A., Nikitin S., Terziyan V., Veijalainen J., How Does UDDI Fit Semantic Needs, In: M. Bramer and V. Terziyan (Eds.): *Industrial Applications of Semantic Web, Proceedings of the 1-st International IFIP/WG12.5 Working Conference IASW-2005*, August 25-27, 2005, Jyvaskyla, Finland, Springer, IFIP, pp. 141-159.

40. Salmenjoki K., Tsaruk Y. , Arumugam G., Information Architectures for Semantic Web Applications, In: M. Bramer and V. Terziyan (Eds.):*Industrial Applications of Semantic Web, Proceedings of the 1-st International IFIP/WG12.5 Working Conference IASW-2005* , August 25-27, 2005, Jyvaskyla, Finland, Springer, ISBN:038-72856-8-7, pp. 247-260.

41. Kaykova O., Khriyenko O., Terziyan V., Zharko A., RGBDF: Resource Goal and Behaviour Description Framework, In: M. Bramer and V. Terziyan (Eds.): *Industrial Applications of Semantic Web, Proceedings of the 1-st International IFIP/WG12.5 Working Conference IASW-2005*, August 25-27, 2005, Jyvaskyla, Finland, Springer IFIP, pp. 83-99.

42. Nikitin S., Terziyan V., Tsaruk Y., Zharko A., Querying Dynamic and Context-Sensitive Metadata in Semantic Web, In: *Proceedings of the International Workshop on Autonomous Intelligent Systems – Agents and Data Mining (AIS-ADM-05)*, June 6-8, 2005, St. Petersburg, Russia, Springer, LNAI 3505, pp. 200-214.

43. Terziyan V., SmartResource: Utilizing Semantic Web Services to Monitor Industrial Devices, In: A. Heimburger, T. Kauppinen, H. Lehtinen, J. Multisilta, M. Paajanen and K. Rytkonen (eds.): *XML - the Enabling Technology for Integrating Business Processes,* 2005, Tampere University of Technology, Pori, Finland, ISBN 952-15-1312-8, ISSN 1795-2166, pp. 89-103.

44. Terziyan V., SmartResource: Utilizing Semantic Web Services to Monitor Industrial Devices, In: A. Heimburger, T. Kauppinen, H. Lehtinen, J. Multisilta, M. Paajanen and K. Rytkonen (eds.): *XML - the Enabling Technology for Integrating Business Processes,* 2005, Tampere University of Technology, Pori, Finland, ISBN 952-15-1312-8, ISSN 1795-2166, pp. 89-103.

45. Salmenjoki K., Terziyan V., Approaching Web-Based Expertise with Semantic Web, In: G. Papadourakis and I. Lazaridis (eds*.): New Horizons in Industry, Business and Education, Proceedings of the 4-th International Conference NHIBE-2005*, August 25-26, 2005, Corfu, Greece, pp. 310-315.

46. Terziyan V., Kaykova O., Vitko O., Titova L., Ontology-Based International Degree Recognition, In: Information Technologies in *Education, Proceedings of the ITE-2005 Workshop*, Yerevan, Armenia, 13-17 June, 2005, Publications of IEEE Computer Society Technical Committee on Learning Technology, ISSN 1438-0625.

47. Keleberda I., Lesna N., Makovetskiy S., Terziyan V., Personalized Distance Learning based on Multiagent Ontological System, In: E. Kinshuk et al (eds.), Proceedings *of the 4-th IEEE International Conference on Advanced Learning Technologies,* Joensuu, Finland, August 30 – September 1, 2004, IEEE Computer Society Press, pp. 777-779.

48. Terziyan V., Vitko O., Learning Bayesian Metanetworks from Data with Multilevel Uncertainty, In: Max Bramer and Vladan Devedzic (eds.), *Proceedings of the First International Conference on Artificial Intelligence and Innovations (IFIP 18-th World Computer Congress),* Toulouse, France, August 22-27, 2004, Kluwer Academic Publishers, pp. 187-196.

49. 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.

50. Kaykova O., Kononenko O., Terziyan V., Zharko A., Community Formation Scenarios in Peer-to-Peer Web Service Environments, In: *Proceedings of the IASTED International Conference on Databases and Applications (DBA 2004),* February 17-19, 2004, Innsbruck, Austria, ACTA Press, ISBN: 0-88986-383-0, ISSN: 1027-2666, pp. 62-67.

51. Khriyenko O., Kononenko O., Terziyan V., OntoEnvironment: An Integration Infrastructure for Distributed Heterogeneous Resources, In: *Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN 2004),* February 17-19, 2004, Innsbruck, Austria, ACTA Press, ISBN: 0-88986-385-7, ISSN: 1027-2666, pp. 319-324.

52. Kaykova O., Khriyenko O., Kovalainen M., Zharko A., Visual Interface for Adaptation of Data Sources to Semantic Web, In: *"IASTED International Conference on Software Engineering (SE 2004)",* Innsbruck, Austria, 17-19 February, 2004.

53. Terziyan V., Semantic Web Services for Smart Devices Based on Mobile Agents, In: J. Marx Gomez (ed.), *Intelligent Mobile Agents in Peer-to-Peer Networks: Proceedings of the Forth International ICSC Symposium on Engineering Intelligent Systems (EIS-2004),* Madeira, Portugal, February 29 – March 2, 2004, ICSC Academic Press, Canada, ISBN: 3-906454-35-5, pp. 10-16.

54. 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: OTM 2003 Workshops, Lecture Notes in Computer Science,* Vol. 2889, Springer-Verlag, 2003, pp.279-291.

55. 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.

56. Jonker C., Terziyan V., Treur J., Temporal and Spatial Analysis to Personalize an Agent's Dynamic Belief, Desire and Intention Profiles, In: M. Klush et al. (eds.), *Cooperative Information Agents VII: Proceedings of the 7-th International Workshop on Cooperative Information Agents (CIA-2003),* Helsinki, Finland, August 27-29, 2003, Lecture Notes in Artificial Intelligence, V. 2782, Springer-Verlag, pp. 289-315.

57. Terziyan V., Ryabov V., Abstract Diagnostics Based on Uncertain Temporal Scenarios, In: M. Mohammadian (ed.), *Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA-2003),* February 12-14, 2003, ISBN 1740880684, Vienna, Austria, pp. 327-337.

58. Ryabov V., Terziyan V., Industrial Diagnostics Using Algebra of Uncertain Temporal Relations, In*: Proceedings of the 21-st IASTED International Multi-Conference on Applied Informatics (AI-2003),* February 10-13, 2003, Innsbruck, Austria, ACTA Press, ISBN 0-88986-341-5, ISSN 1027-2666, pp.351-356.

Scientific Project Reports

59. Katasonov A., Khriyenko O., Nikitin S., Terziyan V., SmartResource Platform for Web-Service Interaction's Semantic Log, *Technical Report (Deliverable D 3.3), SmartResource Tekes Project,* Agora Center, University of Jyvaskyla, March-October, 2006.

60. Katasonov A., Khriyenko O., Terziyan V., Tsaruk Y., SmartResource Platform in Distributed Power Networks Maintenance, *Technical Report (Deliverable D 3.2), SmartResource Tekes Project,* Agora Center, University of Jyvaskyla, March-October, 2006.

61. Kaykova O., Khriyenko O., Terziyan V., General Networking Framework, *Technical Report (Deliverable D 3.1), SmartResource Tekes Project,* Agora Center, University of Jyvaskyla, January-May, 2006.

62. Pulkkinen M., Naumenko A., Nieminen K. Metso Paper Oy and Trusteq Oy, *Case Report*, *Mobile Design Patterns and Architectures Research Project,* Information Technology Research Institute, University of Jyvaskyla, April, 2006.

63. Terziyan V. et al (Industrial Ontologies Group), SmartResource Annual 2005 Report, *Technical Report (annual report), SmartResource Tekes Project,* Agora Center, University of Jyvaskyla, 2005.

64. Mazhelis O. and Naumenko A. Security Patterns in Software Application Design, *Technical Report #10, Mobile Design Patterns and Architectures Research Project,* Information Technology Research Institute, University of Jyvaskyla, October, 2005.

65. Naumenko A., Nikitin S., Terziyan V., Zharko A., SmartResource Prototype Environment, Version 2.0, *Technical Report (Deliverable D 2.3), SmartResource Tekes Project,* Agora Center, University of Jyvaskyla, August-December, 2005.

66. Tiitinen P., Naumenko A., Nikitin S., Bartholdt J., Toma I., Roman D., Tausch B., Requirements Analysis on Service Registries, *Deliverable D2.II-1, Adaptive Services Grid FP6 – 004617*, August 2005.

67. Kaykova O., Khriyenko O., Terziyan V., Zharko A., Design of the SmartResource Platform, *Technical Report (Deliverable D 2.2), SmartResource Tekes Project,* Agora Center, University of Jyvaskyla, June-October, 2005.

68. Kaykova O., Khriyenko O., Terziyan V., Zharko A., General Proactivity Framework (Pro-GAF), *Technical Report (Deliverable D 2.1), SmartResource Tekes Project,* Agora Center, University of Jyvaskyla, February-May, 2005.

69. Terziyan V. et al (Industrial Ontologies Group), SmartResource Annual 2004 Report*, Technical Report (annual report), SmartResource Tekes Project,* Agora Center, University of Jyvaskyla, 2004.

70. Kaykova O., Khriyenko O., Kovtun D., Marttinen J., Naumenko A., Terziyan V., Tsaruk Y., Zharko A., SmartResource Prototype Environment, Version 1.0, "Adaptation Stage", *Technical Report (Deliverable D 1.3), SmartResource Tekes Project,* Agora Center, University of Jyvaskyla, July-December, 2004.

71. Kaykova O., Khriyenko O., Kovtun D., Marttinen J., Naumenko A., Terziyan V., Tsaruk Y., Zharko A., General Adaptation Framework: Framework for Semantic Adaptation of Maintenance Resources, *Technical Report (Deliverable D 1.2), SmartResource Tekes Project,* Agora Center, University of Jyvaskyla, July-October, 2004.

72. Kaykova O., Khriyenko O., Naumenko A., Terziyan V., Zharko A., Resource State/Condition Description Framework (RscDF), *Technical Report (Deliverable D 1.1), SmartResource Tekes Project,* Agora Center, University of Jyvaskyla, June-August, 2004.

Academic Degree Theses

73. Zhovtobryukh D., Context-aware Web Service Composition, *PhD Thesis*, Department of Mathematical Information Technology, University of Jyväskylä, December 2006.

74. Katasonov A., Dependability Aspects in the Development and Provision of Location-Based Services, *PhD Thesis*, Department of Computer Science and Information Systems, University of Jyväskylä, April 2006.

75. Tsaruk Y., Information Base for Representation of Management Information in DIFFSERV , *M.Sc. Thesis,* Mathematical Information Technology Department, University of Jyväskylä, June 2005.

76. Kovtun D., Mobile Services for Wireless Private Area Networks based on Ad Hoc Connectivity, *M.Sc. Thesis*, Mathematical Information Technology Department, University of Jyväskylä, December 2004.

77. Naumenko A., Analysis and Semantic Description of Role Based Access Control Models, *M.Sc. Thesis,* Mathematical Information Technology Department, University of Jyväskylä, October 2004.

78. Nikitin S., Web-Service for Computational Error Estimation, *M.Sc. Thesis,* Mathematical Information Technology Department, University of Jyväskylä, August 2004.

79. Khriyenko O., Distributed Mobile Web Services based on Semantic Web , *M.Sc. Thesis*, Mathematical Information Technology Department, University of Jyväskylä, October 2003.

80. Kononenko O., Ontological Support for Industrial Maintenance of Smart-Devices , *M.Sc. Thesis*, Mathematical Information Technology Department, University of Jyväskylä, October 2003.

81. Zharko A., Peer-to-Peer Ontological Discovery of Mobile Service Components in Semantic Web , *M.Sc. Thesis,* Mathematical Information Technology Department, University of Jyväskylä, October 2003.

SmartResource

Project Budget

Project Budget

	1st Year	2nd Year	3rd Year	Total (Euro)
Salary & Overheads:	187 300	194 000	289 000	670 300
Travels:	9 000	4 500	10 000	23 500
Equipment & Consumables:	3 700	1 500	1 000	6 200
TOTAL (Euro):	**200 000**	**200 000**	**300 000**	**700 000**
Funding from companies & TEKES:				
Metso Automation:	10 000	10 000	20 000	40 000
TeliaSonera:	20 000	20 000	20 000	60 000
TietoEnator:	10 000	10 000		20 000
JSP:	own work	own work	own work	own work
ABB:			20 000	20 000
TEKES:	160 000	160 000	240 000	560 000