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

Vagan Terziyan

Industrial Ontologies Group, University of Jyvaskyla (Finland)
vagan@cc.jyu.fi

Abstract

The paper summarizes research findings related to SmartResource project (2004-2007) funded by Tekes and industrial companies. The main objectives was research and development of the large-scale distributed environment for integration of smart devices, web services and humans based on combination of Semantic Web, agent technologies and service-oriented architecture. A prototype platform for self-maintained smart resources in smart spaces 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 related industrial case study. Several lessons have been learned during the project in addition to published results, which we are going to share with scientific community. We also present a vision how to utilize project results to design various complex smart spaces taking into account such issues as interoperability, coordination, self-manageability, reputation and trust in future generation smart space environments.

1. Introduction

Being predominantly small-scale, specialized and often isolated, smart space environments are penetrating into our life very fast. They gracefully weave themselves into surrounding physical infrastructure to bring specific non-obtrusive value-added functionality to humans and thereby act as their invisible servants carrying out these functions largely autonomously and in the background. Aside from apparent but complicated challenge of value-added autonomous functionality design, there is a fundamental challenge for seamless device interoperability which must be solved in the first place, as devices form the backbone of any smart space. This interoperability quest is also identified as crucial in many adjacent science and technology areas such as future Internet and flexible service architectures. However, it receives key importance and a rather comprehensive view particularly within the smart space research and development area. Specifically, we see at least the following two interoperability problems: interoperability between the devices produced and programmed by different vendors and/or providers, and the need for seamless and flexible collaboration (including discovery, coordination, conflict resolution and possibly even negotiation) amongst the smart space devices and services. To tackle these problems utilization of Semantic Web languages and technologies for declarative specification of devices' and services' behavior, application of software agents as engines executing those specifications and establishment of common ontologies to facilitate and govern seamless interoperation of devices within smart spaces seem to be crucially important.

The *SmartResource* project [1] started in 2004 and funded by Tekes [2] and industrial companies (Metso Automation, TeliaSonera, TietoEnator, ABB) has officially ended in April 2007. Its objectives were research and development of the large-scale environment for integration of smart devices, web services and humans based on Semantic Web and agent technologies. A prototype platform has been designed and implemented for particular tasks of industrial partners. The project belongs to the Industrial Ontologies Group[3] research roadmap towards the *Global Understanding Environment* (GUN) [14, 13, 4]. When applying Semantic Web in the domains of ubiquitous computing and smart spaces, 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.

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.

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 architecture and main concept behind it can be seen in Figure 1.

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 (e.g. XML to RDF). 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 and Multi-Agent Systems (MAS) 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.

---

[1] Web pages of SmartResource project: http://www.cs.jyu.fi/ai/OntoGroup/SmartResource_details.htm
[2] Finnish Funding Agency for Technology and Innovation: http://www.tekes.fi/eng/
[3] Web pages of Industrial Ontologies Group: http://www.cs.jyu.fi/ai/OntoGroup/index.html

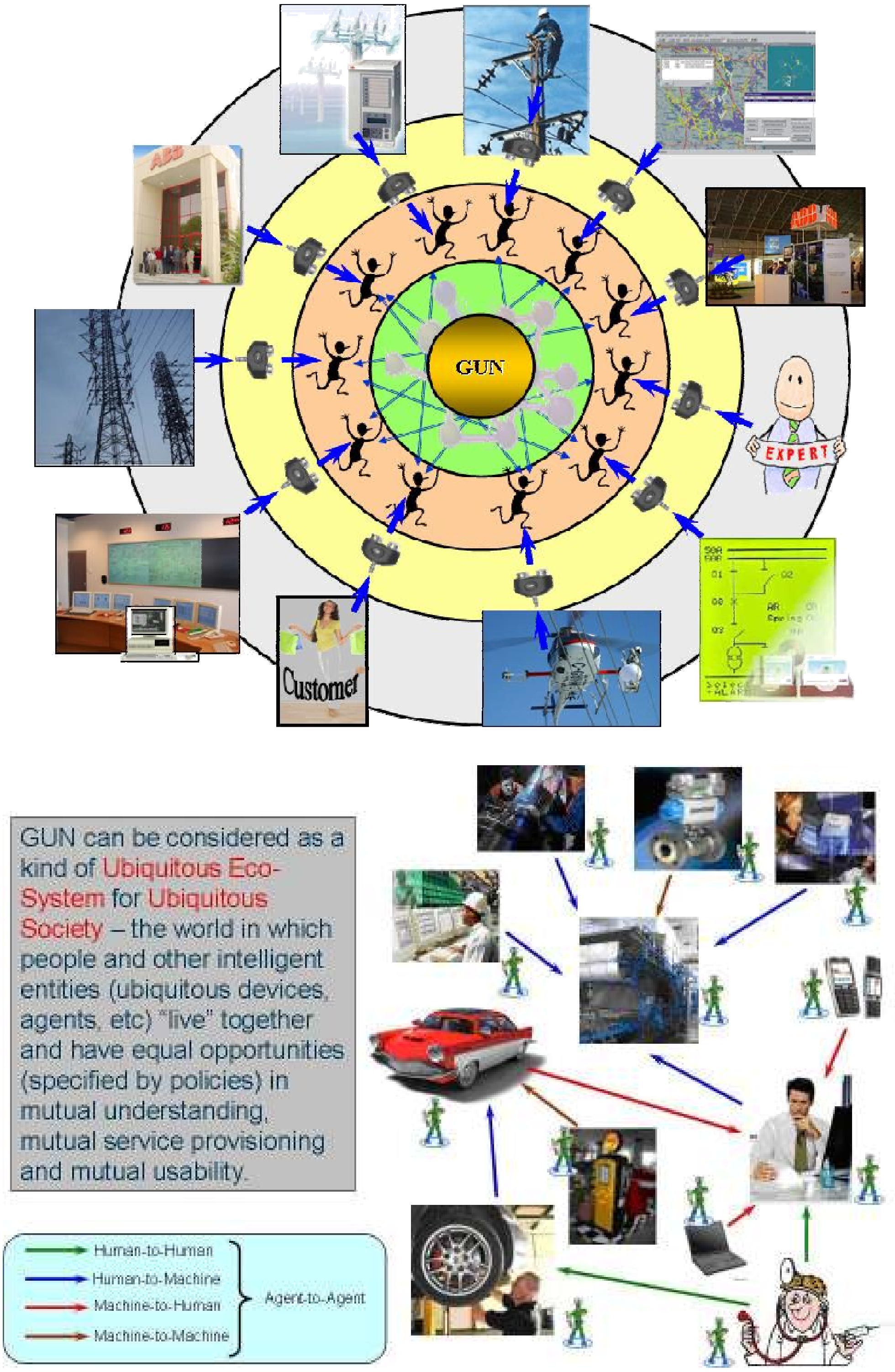

Figure 1. The concept of a Global Understanding Environment

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 engineering (SE) technologies, which are integral parts of it, e.g. object-oriented SE, service-oriented architectures, 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.

We consider resources to be smart if they are Web-accessible, proactive, self descriptive and self managed. What are “resources” however? Semantic Web community mainly 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 and Web-services, data, machines and devices, humans, organizations, communication systems, protocols and networks, processes, concepts, models, ontologies, messages, standards, etc.

In this paper we summarize the basic scientific challenges and achievements of SmartResource project as separate chapters (Chapter 2 describes framework for designing adapters to smart resources; Chapter 3 describes the way to model proactivity of the resources; Chapter 4 introduces approach on how to model collaborative behavior of the resources; Chapter 5 summarizes evolution of RDF needed to meet GUN requirements; Chapter 6 presents GUN platform architecture and results of the industrial case studies made with GUN pilot implementation; Chapter 7 briefly summarizes related work) and providing vision (Chapter 8) on how to utilize GUN as agent-driven semantic middleware platform for device interoperability in various smart space environments. Some conclusions are provided in Chapter 9.

2. 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 2). 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 [5].

RscDF extends RDF by making it more suitable for semantic annotation of dynamic and context-sensitive data about the resources. It provides opportunity to put any RDF statement into context, which is described by a container of RDF statements. Appropriate schema also includes some specific properties able to describe dynamic and if needed multilayered context of statements

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 and a smart way how to automatically design an adapter for some resource by combining existing components based on the resource semantic description. In [4] 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 built from hardware, software and even "human" components.

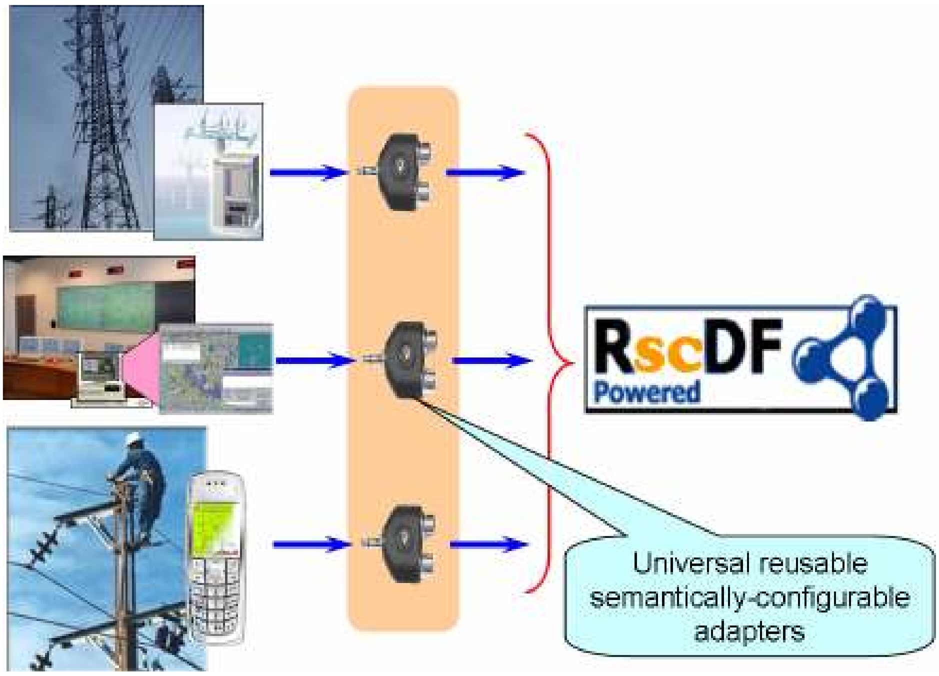

Figure 2. The challenge of universal self-configurable adapters

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

3. 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 3). 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 [6]. The extension allows making explicit links from behavioral properties of proactive resources to appropriate atomic software components, which are intended to implement described behavior when appropriate.

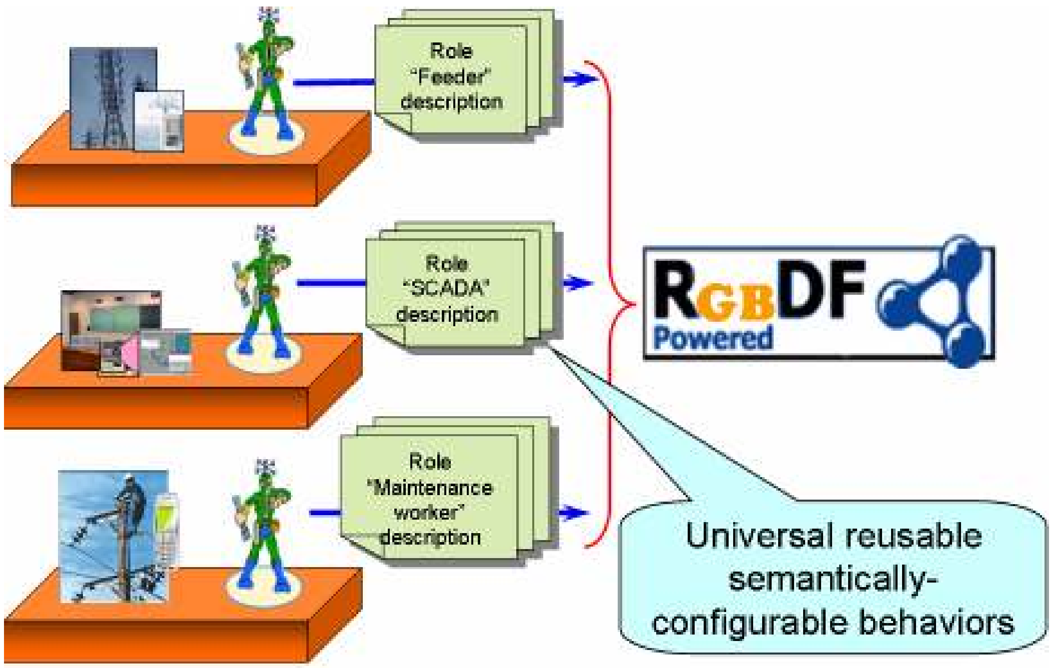

Figure 3. The challenge of universal self-configurable behaviors

Consider the example scenario in Figure 4. 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 its 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.

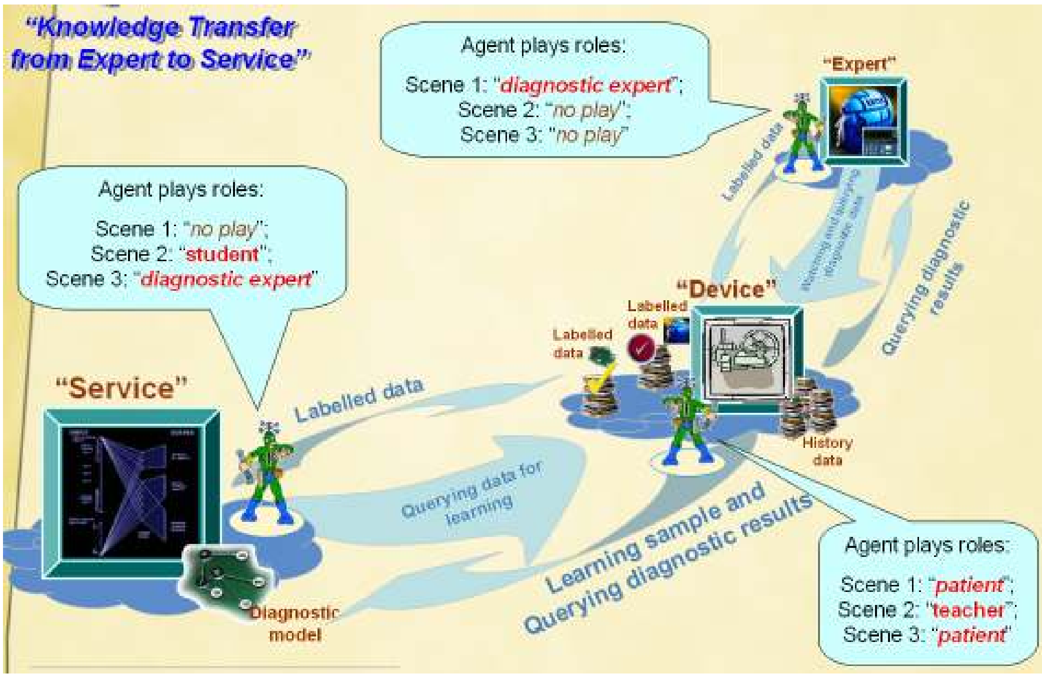

Figure 4. Sample scenario in which agents can change own roles depending on the context

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.

4. 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. In SmartResource project we designed RpiDF (Resource Process/Integration Description Framework) as a tool for semantic annotation of policies and metarules for controlling individual behaviors of the resources towards achieving collaborative goals. RpiDF extends RDF by making it more suitable for semantic annotation of collaborative behavior of the resources. The extension allows putting explicit constrains on individual rules, plans and utilized atomic behavioral software components, which are intended to implement corroborative goal-driven behaviors (scenarios) of the group of proactive resources (systems). General Networking Framework (GNF) should provide ontologies and tools to design, share, reuse and integrate universal semantically-configurable scenarios for required coordination (see Figure 5).

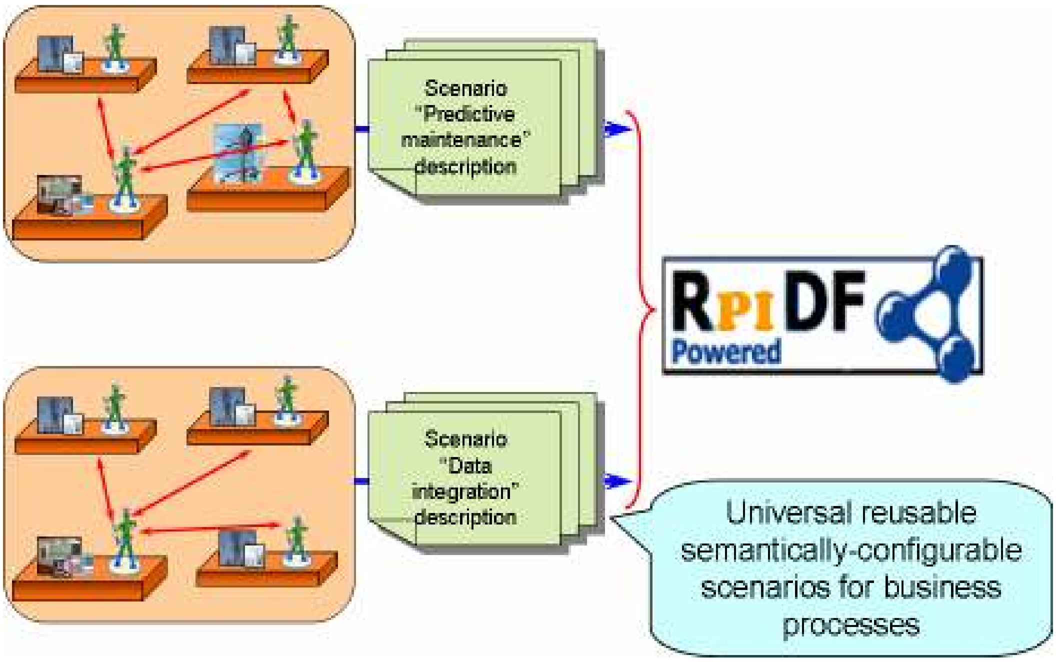

Figure 5. The challenge of universal self-configurable scenarios for coordination

GNF is also a technology and a platform for integrating 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 6). Blue (one-way directional) arrows in Figure 6 represent such part-of hierarchy of resources and red (two-way directional) arrows shows show possible communication links between corresponding agents.

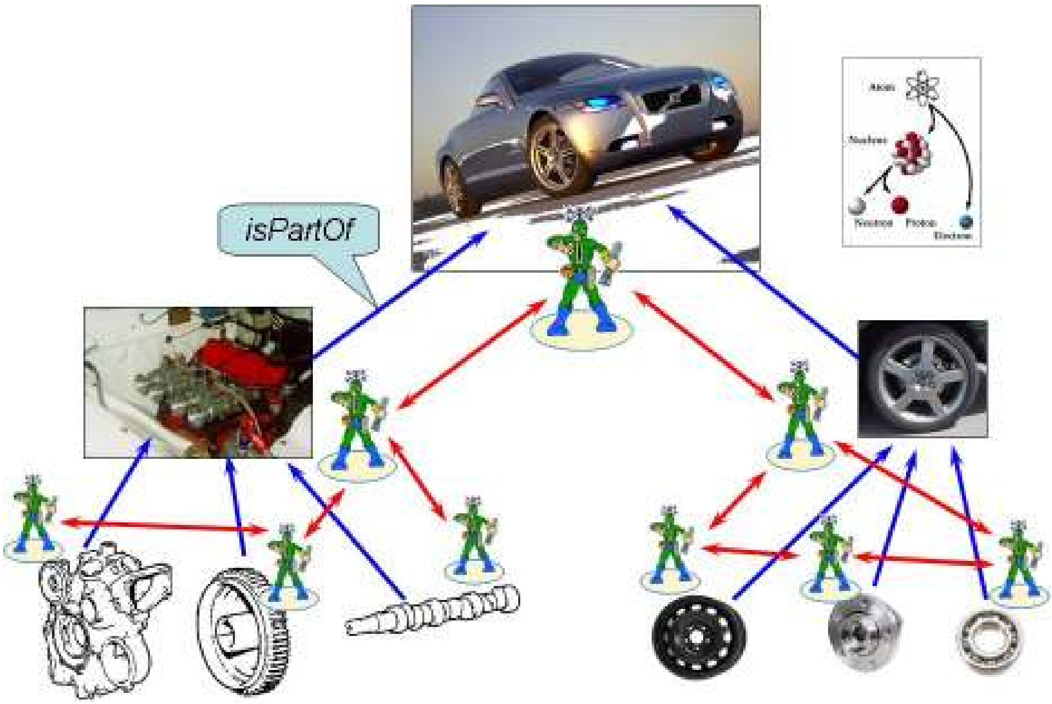

Figure 6. Example of a "part-of" hierarchy of resources, which resulted to corresponding hierarchy of agents

Another important concern is: What is a process in GUN environment? Consider the following two axioms of GUN (see also Figure 9 later in the text):

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.

By the "Industrial World" in the above definitions we mean such part of the more general World of Things or Internet of Things (see e.g. report of the International Telecommunication Union in: http://www.itu.int/osg/spu/publications/internetofthings/), which contains only resources that are taking part in various industrial processes. "Resource agent" is such an agent, who takes care of certain recourse of the Industrial World (according to the basic GUN vision in Figure 1).

As all of the 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, whose resource, to be taken care of, is the Industrial World as whole. Such agent will be on the top of the hierarchy of resource agents.

Each industrial resource can theoretically be involved in 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 7).

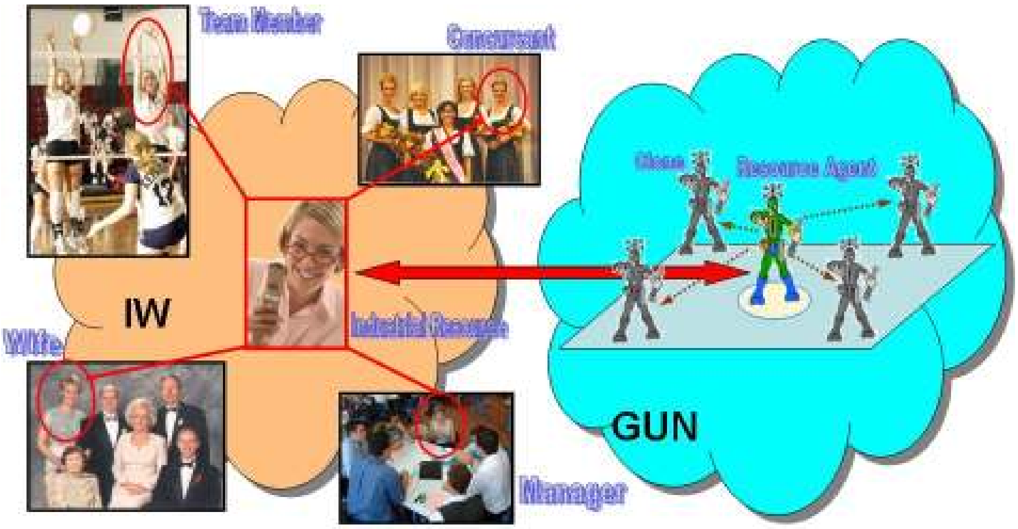

Figure 7. Multiple roles of a resource in the Industrial World and appropriate agent-clones in GUN

Each industrial resource, which joins some commitment, will behave according to restrictions the rules (policies) of that commitment require. The more commitments individual resource takes, the more restriction will be put on its behavior (see Figure 8). General types of possible policies used by organizations to manage (i.e. restrict freedom of choice in behavior) individuals include:

- Instructions (e.g. "drink at least 2 liters of water every day");
- Conditional Instructions ("whenever hear alarm, call security");
- Commitments (e.g. "promise to love your spouse forever");
- Conditional Commitments (e.g. "promise to take care of partner in case of illness");
- Restrictions (e.g. "no smoking");
- Conditional Restrictions ("do not use elevator in case of fire").

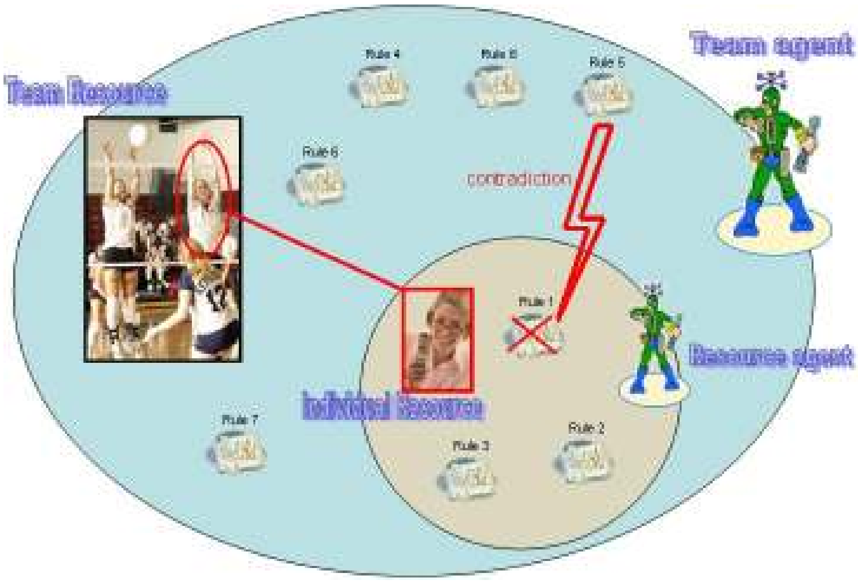

Figure 8. Individual vs. team resource freedom

Concerning the opposite (i.e. how individual behavior of an agent can change policies of the community), this will require from the agent to initiate the process of negotiating with the community on appropriate change in shared policies and if all (or in some cases the majority) will accept this offer then the change might take place. Otherwise the agent will have to accept valid restrictions of appropriate community or quit this community and join another one where agreed collaborative restrictions seem to be more appropriate.

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 9).

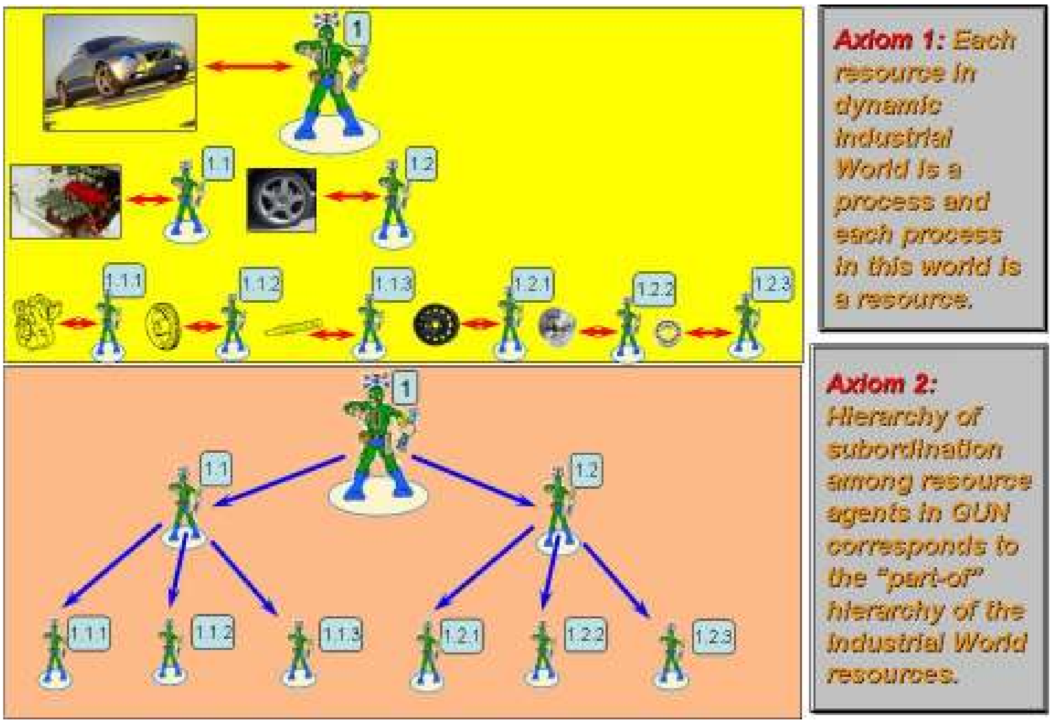

Figure 9. Agent subordination according to GUN axioms

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. After that resources can be considered as 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 making dynamic (smart) routing, where a smart message itself (not the nodes) decides where to go further within a network and the message is also able to collect own history and communicate with other messages.

5. RDF Evolution towards Context Description Framework

Based on research described in Chapters 2-4, we can conclude that RDF as such is not enough 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 10).

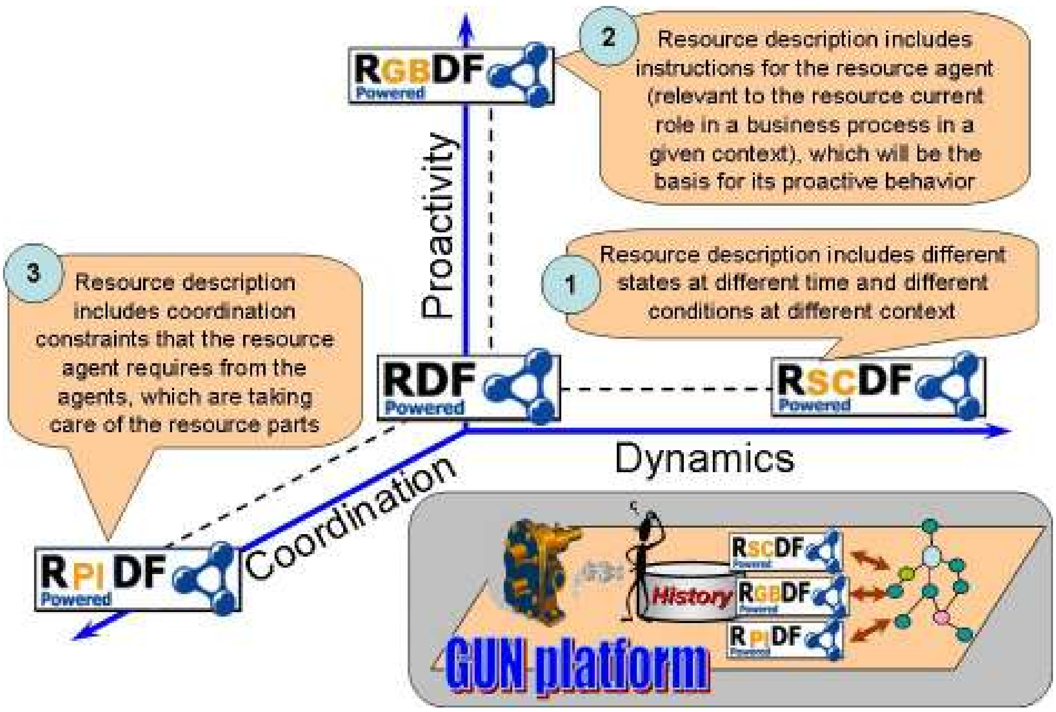

Figure 10. Three dimensions of developing RDF towards the Internet of Things domain

The integrated view to all three extensions (RscDF, RgbDF and RpiDF) is based on the same framework, called Context Description Framework (CDF) (see [8]) 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.

A Context Description Framework is meant to model the context dependence of the world properties. It allows us make two significant steps in the resource description approach. In CDF logically proceeded 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 as shown in Figure 11).

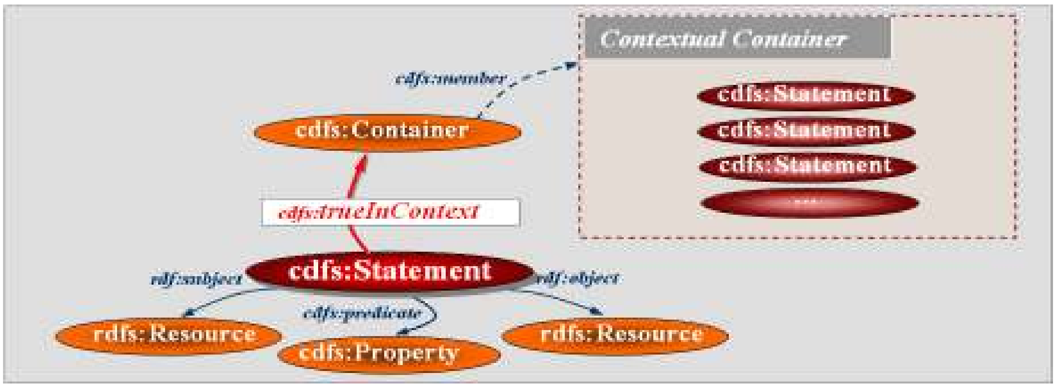

Figure 11. A quadruple vision of the statement

The summary of three described frameworks and the appropriate conceptual difference between RscDF, RgbDF and RpiDF are shown in Figure 12.

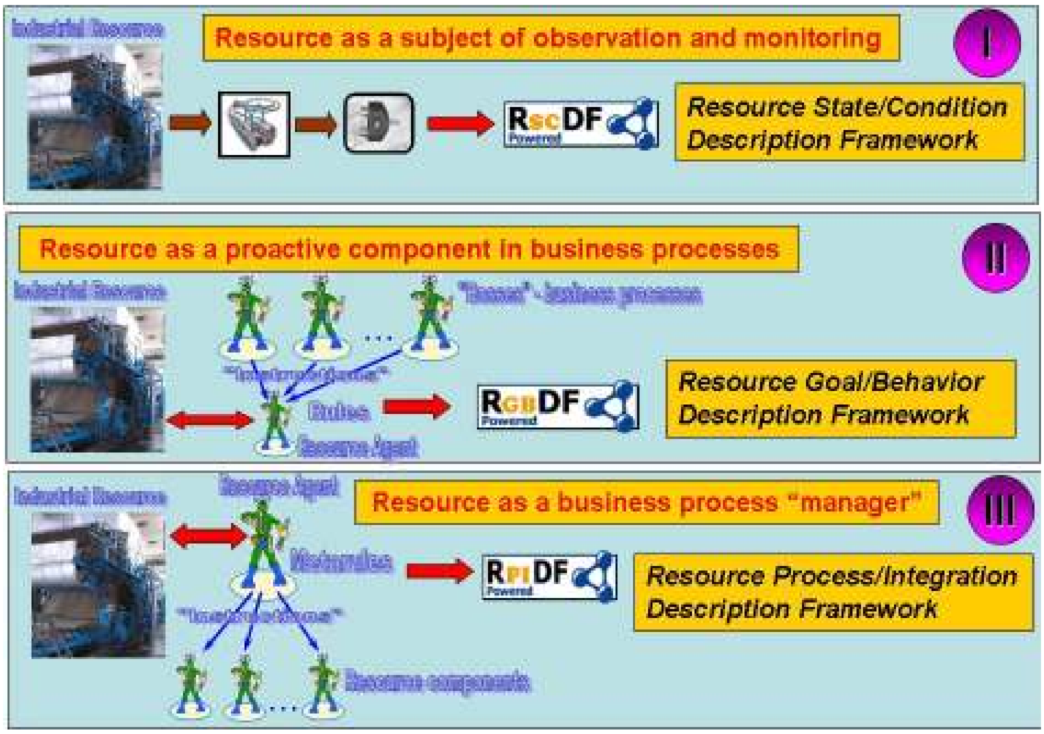

Figure 12. The conceptual difference between three frameworks in SmartResource project

As it was mentioned above, the GUN environment within the scope of SmartResource project was 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.

6. GUN Platform Architecture

In GUN platform architecture we consider agent platform components to be agents themselves. In Figure 13, the 3-layered GUN platform for a particular industrial resource management is shown.

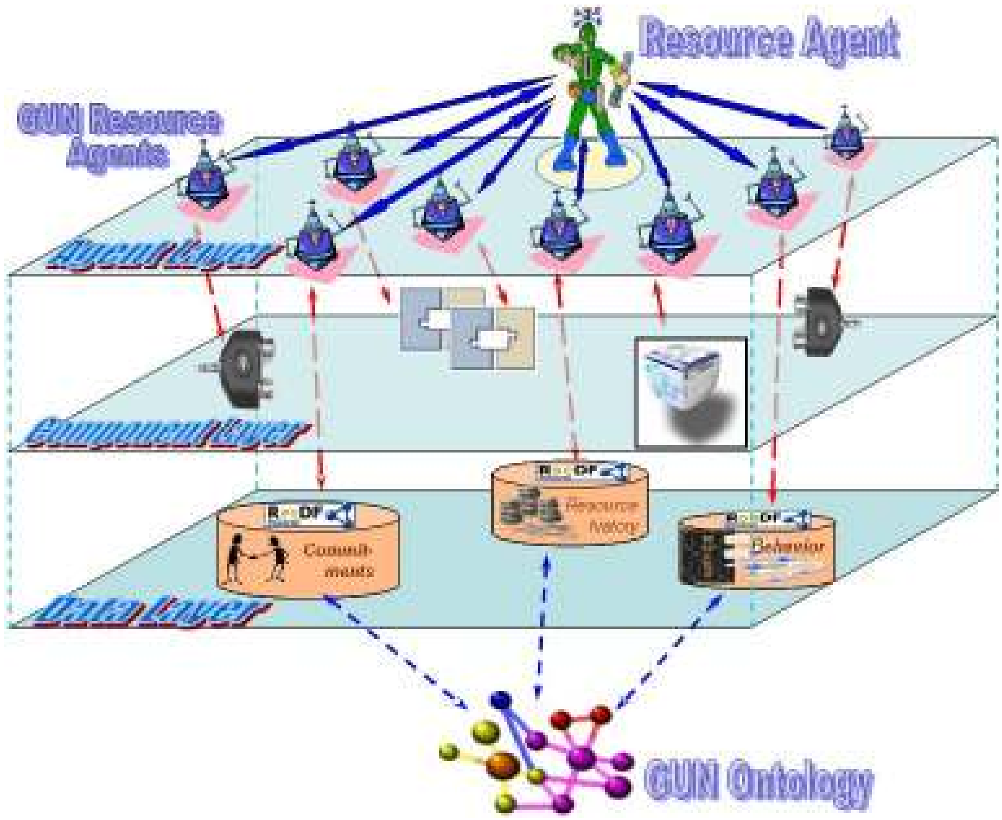

Figure 13. A 3-layered GUN platform for managing smart industrial resource

The Agent Layer in Figure 13 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 this component whenever needed; or the agent 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 (on the own GUN agent-platform) all needed adapters, histories, behavior sets, software components, commitments and reusable coordination patterns (in RpiDF) and other GUN resources. 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.

During SmartResource project the pilot version of GUN platform has been utilized and tested within two industrial case studies. First case study of GUN pilot tools for paper industry has been made in cooperation with Metso Automation (see e.g. [10]). Study was related to semantic management of fault data coming to the Metso Maintenance Center from client paper machines. The designed GUN-based system provided 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. 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. Another industrial case study made in collaboration with ABB Distribution Automation (see e.g. [12]) is related to the domain of distributed power network maintenance. Study was related to the possibility to utilize GUN platform for 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 to be used for e.g. risk analysis, facilitation of fault localization, operator interface enhancement, etc.

7. Related Work

Recent advances in networking, sensor and RFID technologies allow connecting various physical world objects to the information and communication infrastructure, which could, ultimately, enable realization of the "Internet of Things", the Ubiquitous Computing and Smart Spaces visions. Such interconnectivity of computing and physical systems could, however, become the "nightmare of ubiquitous computing" [7], 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. The IBM vision of autonomic computing 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.

According to [2], 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. 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*[4]. 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. 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 [1] 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 [11]. 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 a nutshell, next generation Semantic Web systems will necessarily have to deal with the increased heterogeneity of semantic sources [9], which partly corresponds to the trends related to the Internet of Things roadmap for the future development [2].

According to [3], a middleware for future smart spaces in general and for smart homes in particular should have the following layered architecture: physical layer (various devices and appliances); sensor platform layer (middleware communication with physical layer and uniform representation of data coming from physical layer); service layer (maintains available basic, composite and standard services); knowledge layer (ontology of the various services with reasoning engine on top of it); context management layer (contexts of interest able to restrict service activation for various applications and context engine); and application layer (application manager to activate and deactivate services and smart spaces' design and simulation environment). As can be seen from above, the GUN architecture has all such architectural components and in addition to it GUN has also an embedded agent-driven proactivity and self-management.

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". If the Future Internet will allow more natural integration of sensor networks with the rest of the Internet, the amount and heterogeneity of resources in the Web

[4] David Brock and Ed Schuster (MIT Data Center) at *Semantic Days 2006*, Norway, April 26, 2006, http://www.olf.no/english/news/?30357

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

8. Discussion: GUN as Future Generation Smart Space Middleware

The future generation smart spaces can be considered as fully interoperable (though heterogeneous), highly dynamic and extensible environments. To achieve that, a specialized agent-driven middleware platform such as GUN seems to be a reasonable option. It is envisioned that each ubiquitous smart device (as well as each individual service exposed as an individually accessible entity through the environment) will be assigned a representative agent within GUN. The resulting multi-agent system will be exploited as a mediation facility enabling rich cooperation capabilities (e.g., discovery, coordination, adaptability, and negotiation) amongst the devices inhabiting the smart space environment. Utilization of semantic technologies in GUN ensures efficient and autonomous coordination among the agents and will thus ensure interoperability between associated devices and services. Also appropriate ontologies should be designed as an important asset contributing to interoperability realization within future smart space environments. These ontologies will be used not only for the benefit of GUN middleware architecture, but also and most importantly for facilitation of interoperability and integration of existing and brand-new future devices, services and methodologies which will be later developed within the smart spaces domain. Appropriate declarative specification of smart space components' behavior and using sophisticated choreographic control of agents in an infrastructureless networked environment, the GUN-based middleware will enable various devices and services to automatically discover each other and to configure complex services functionally composed of the individual services' and devices' functionalities.

Further steps towards utilization of GUN approach in smart spaces domain include:

1. Design of the Upper Ontology for the domain of smart space environments and Device, Service and User extendable ontologies.
2. Design of adapters for linking smart space devices and services to associated agents within the GUN platform.
3. Design of appropriate scenarios for self-management, self-configuration and integration of the resources in smart spaces, including cross-layer scenarios (amongst devices, services and users).
4. Design of AI support tools for GUN platform allowing agents to automatically create and utilize configuration plans (in addition to manually predefined), to learn (data mining, knowledge discovery and utilization for management of the underlying networking architecture) and improve individual and collective performance accordingly based on observed histories of the managed resources.
5. Design of specialized agent-driven protocols for efficient discovery of the resources (devices and services) in ad-hoc peer-to-peer smart space environments.
6. Design of support mechanisms for agent-based service composition planning, assembling and deploying in a smart space environment.
7. Design of support mechanisms for user-driven on-demand service selection and/or composition in a smart space environment.

The GUN concept apparently entails a vision of a multifaceted, multi-purpose and multipronged middleware platform applying multidisciplinary approach to extension and enhancement of the future smart space ecosystem vision. The GUN platform should be rather seen as a meta-structure on top of the future smart space ecosystem or as intelligent stratum between the smart space device layer and the future service oriented environments.

The important elements of GUN are also the proactive multi-agent semantic coordination and management system and context-aware, user-driven framework for automated composition of reconfigurable services. This particular middleware solution adds to overall network system flexibility, openness, (re-)configurability and manageability.

8.1. Interoperability in smart spaces

By proclaiming interoperability as its major ultimate objective, GUN approach deals with four major types of interoperability problem: technical interoperability (being the capability of devices, protocols and other technical standards to co-exist and interoperate), semantic interoperability (being the capability of various system components to treat and interpret exchanged data and information identically and share a common understanding of it), pragmatic interoperability (being the capability of system components to capture willingness of partners to collaborate or, more generally, to capture their intentions) and cross-layer interoperability (a collaboration capability, which involves resources from different layers of a system).

Technical interoperability will be achieved through the agent-based mediation between different devices and standards with the aid of special adapter components and tunneling mechanisms. Semantic interoperability is the main focus of the GUN approach as it is a prerequisite for seamless information internetworking and integration, and for smooth autonomous communication between various resources within a smart space environment.

Semantic interoperability can be achieved by exploitation of rich metadata describing informational objects and semantic resource descriptions written in compliance with well-established semantic standards and on the base of predefined domain ontologies.

Pragmatic interoperability amongst smart space components will be achieved through appropriate design of declarative specifications of such components' behavior and on-the-fly agent-based identification of this behavior using given descriptions.

Finally, the most innovative type of interoperability GUN intends to natively provide is the so-called *cross-layer interoperability*, e.g., interoperability between devices and services in a smart space environment. This particular class of interoperability problems is often difficult to solve even on individual basis. However, GUN provides native support for cross-layer interoperation by implementing the paradigm of resource-oriented networking. This paradigm enforces unified treatment of various system components, e.g., devices, services, applications and even users, as different types of resources. The communication is then established between resources regardless their particular type provided that negotiation is performed by resources' representing agents and specific flexible standards for unified resource description are used.

8.2. Flexible coordination in smart spaces

As smart space environments are basically deployed to provide users with localized, customized, value-added and autonomously operating services, GUN targets such service creation and provisioning framework that would emphasize the above mentioned characteristics of ubiquitous services. Customization, personalization, added value, dynamicity and autonomy of services is to be achieved through construction and utilization of context-aware, adaptable and reconfigurable composite service networks. Service networks can be composed using declarative specifications of service models. Reconfigurability of service networks is made possible via utilization of hierarchical modeling of service control and its run-time execution. Dynamic adaptation of services is performed by special context-aware control components built in service networks. The traditional tradeoff "customization vs. autonomy" can be dealt with through a balanced use of user-aware goal-driven on-demand service composition, AI-enriched active context-awareness capturing user intent, and user-collaborative passive context-aware service composition. Though it is a challenging task, utilization of agent-based approach for service composition makes it much more flexible compared to traditional orchestration approaches. Agents can bring many valuable features into a service composition framework, e.g., precomposition, distributed hierarchical control of service networks (not requiring a dedicated underlying infrastructure), and enhanced negotiation of non-functional service parameters. One can easily check that in this particular aspect GUN makes a bridge between smart spaces and service-oriented architectures.

8.3. Self-manageability in smart spaces

GUN brings self-management aboard via presenting highly distributed agent-driven proactive management system. GUN agents monitor various components, resources and properties within the system architecture and react to changes occurred by reconfiguring the architecture in certain way with respect to the predefined configuration plan. Configuration plans basically represent enhanced business models which are adhered to during accomplished communication procedures between different parties. Due to purely distributed layout of the agent system and outstanding agents' programmability, merely all existing and new business models can be formalized and enacted by the GUN management platform. In addition to this, GUN agents are capable of learning via utilizing available data mining algorithms and further dynamically reconfiguring the managed architecture on the basis of acquired knowledge. GUN platform can be deployed on top of any architectural model (including ad-hoc and peer-to-peer) due to benefits of agent technologies and open resource interfaces. Also, GUN platform can make use of contextual information from the managed networking environment.

8.4. Trust and reputation in smart spaces

Trust is identified as one of the major and most crucial challenges of the future computing and communications. We envisage a semantic ontology-based approach to building a universal trust management system. To make trust descriptions interpretable and processable by autonomous trust management procedures and modules, trust data should be given explicit meaning via semantic annotation. Semantic trust concepts and properties will be utilized and interpreted using common trust ontologies. This approach to trust modeling is especially flexible because it allows for various trust models to be utilized throughout the system seamlessly at the same time. Trust information can be incorporated as part of semantic resource descriptions and stored in dedicated places within the GUN platform. Communication and retrieval of trust information will be accomplished through corresponding agent-to-agent communication. Agents representing communicating resources must be configured appropriately to handle all necessary trust management activities between the corresponding communication parties. Trust management procedures can be realized as a set of specific business scenarios in the form of agent configuration plans.

8.5. Motivating scenario

Consider the following scenario, which can possibly happen in near future:

"Alex is a big tennis fan, who hardly misses a single TV broadcast of a Grand Slam tournament. But he is also a member of a software development team within a big IT enterprise, and this time an important working meeting was scheduled for the time clashing with the TV broadcast of a tournament semi-final match, where Alex's big favorite is playing. Alex was almost desperate because of this disappointing coincidence, but has not given up his hope to see at least a part of the match. Before the meeting, which had to start a little bit earlier than the tennis match, Alex added the match as a concurrent activity to the organizer application within his personal mobile device.

As few minutes before the meeting's start Alex entered the discussion room, where the meeting was to be held, his personal agent (residing in his personal mobile device and managing Alex's PIM applications) recognized that Alex is on the meeting, set presence status to 'at meeting' and switched sounds off (through possible collaboration with GUN device agent).

When the tennis match was about to commence (according to the organizer application information), the personal agent 'realized' it must perform the task of the tennis match broadcast playback as it was set by Alex. However, after sensing the context the agent concluded that neither visual, nor audio playback is possible due to the current Alex's presence status. Livescore visualization with optional textual commentary appeared to be the only acceptable option of the match playback. Thus, the agent acquired an Internet connection through the meeting room's smart space infrastructure (after appropriate trusted negotiation with the corresponding GUN agent) and initiated search procedure to locate appropriate livescore services on the Web. Having found seemingly appropriate services, the agent acquired and processed their semantic service descriptions to ensure that the chosen service is best suited to user, functional and non-functional requirements currently imposed for such a service by the client side. As the agent selected the 'best' livescore service, it then communicated with the corresponding service agent, exchanged trust information, negotiated necessary non-functional parameters, and finally sealed a service contract. As soon as this was done, the livescore service got visualized on Alex's device screen tracking the scoreboard of the started tennis match and its noticeable events. The service session was being controlled by three collaborating agents: Alex's personal agent, device agent and livescore service agent; and could be reconfigured on the fly through appropriate coordination amongst the agents. Now Alex was able to track the match development using his mobile terminal without digressing from the meeting's content.

As soon as the meeting was over, Alex rushed out of the meeting room as he was gripped with desire to see the match with his own eyes. As Alex left the meeting room, his personal agent recognized the meeting was over and realized that the tennis match broadcast became the only current task to fulfill. In the changed context, nothing was restraining Alex from seeing visual broadcast of the match. Through collaboration with GUN agents dwelling in the proximity within the office smart space environment Alex's agent discovered an idle TV wall panel with embedded DVB/DAB receiver in the nearby kitchen. The agent suggested Alex *going there through his personal mobile device. As Alex accepted this suggestion, the agent communicated with the corresponding GUN TV device agent and booked its device for the consecutive hour. When Alex entered the kitchen the TV panel was already switched on and the channel broadcasting the tennis match was selected. Alex made himself comfortable on the kitchen's sofa and soon plunged into watching the match.*

After some minutes of watching the match Alex realized this was going to be a tough encounter, which would likely last for some 2-3 hours, so he decided to go home and watch the rest of the game there. Moreover, he suddenly felt hungry. He entered his organizer application again and scheduled a pizza dinner in 45 minutes at home. Then he stood up and left to his office, where he put on his coat, and finally took an elevator to an underground parking area, where his car was parked. In the meantime, Alex's personal agent did the following: it dropped the TV broadcast service session and released the kitchen TV panel, contacted Alex's office desktop agent to set 'away' presence status and switch the desktop into the sleeping mode, using wireless local area connection got in touch with Alex's car agent to warm up car's engine in advance, queried the available traffic estimator service about current estimate of the duration of Alex's trip from work to home, then it connected with Alex's home smart space agent and scheduled the tennis match TV broadcast in 30 minutes (based on the result the traffic estimator service returned) for the home agent to prepare, and finally it located all available pizzeria services and based on the reputation information received from their corresponding agents it selected the service that was guaranteeing pizza delivery within 45 minutes interval.

When Alex got into his car, which engine was already warmed up, and connected a Bluetooth hands-free device to his mobile terminal, his personal agent offered him to hear an audio broadcast of the progressing tennis match while he is driving (for which the agent found another streaming broadcast service on the Web). Alex accepted the offer and did not miss a key moment of the match until he got to his home. As he entered his apartment, the home smart space agent had already switched on the TV and had set the needed channel. While changing his clothes and washing hands Alex was listening to the broadcast and when he was about to sit and watch the match comfortably pizza delivery arrived. Alex had his dinner and watched the tennis match to an end to celebrate his favorite's hard win and progress to the tournament's final. Owing to autonomous, collaborative and smart behavior of invisible software agents inhabiting the surrounding smart space environments Alex had been able to combine all his conflicting activities with the most convenience and comfort."

Supporting agents' flexibility, proactivity and autonomy to maximize customer's comfort in committing varying and concurrent tasks throughout heterogeneous smart space environments is the major objective of the GUN platform design and exploitation. An important remark is that the described scenario looks sophisticated only on the level of agent communication. From the user point of view it is quite simple as only few actions are required from the user to undertake in order to achieve desired results. Moreover, it is apparent that the course of user actions is significantly simplified in comparison to a routine the user would need to accomplish at present to reach the same goals. No matter how complex agent collaboration patterns could be – they are transparent to the user and are aimed to relieve him/her from significant part of routine actions.

9. Conclusions

Current scientific trends indicate that the research challenges related to interoperability of resources in smart spaces are becoming more and more important. The current state of the art in information and communication technologies puts Semantic Web, Distributed AI (e.g. agents and MAS) and Human-Centric Computing as main candidates for acting as core technologies for building middleware, which will be able to provide such interoperability. This is the main motivation for putting forward our research roadmap towards GUN. One of the most essential results of the SmartResource project (as the first simplest version of GUN) was creation of the “Smart Resource Technology” for designing complex software systems for smart spaces. The technology gets benefits from considering each traditional system component as a “smart resource”, i.e. proactive, agent-driven, self-managing. Such vision opens many new research and business opportunities especially for those resources, which has not been traditionally considered as proactive and self-managed, e.g. mathematical models, communication protocols, ontologies, industrial products, etc.

During SmartResource project as the first stage towards GUN we learned that:

- making adapters from smart space environments to various resources is hard task but can be solved if to design adapters as self-configurable software entities from reusable software components;
- resource dynamics (changing properties and their values over time or depending on a context) is a reality to consider in real applications and semantic technology should be able to provide tools to manage such dynamic descriptions;
- resource proactivity (goal-driven behavior, communication among resources, negotiation, coordination, etc.), which we consider as obligatory to every GUN resource, needs specific semantic tools for markup and management. Reuse of proactivity patterns is also the only way to proceed with heterogeneity and complexity of the resources;
- there is no need to distinguish between objects and processes when talking about resources because in real smart spaces each object is complex and dynamic enough to be considered as a kind of process. The complexity of any resource can be managed with *part_of* hierarchy of its components and by recursively considering each of that components at each level of that hierarchy as autonomous and proactive entity;
- agent technology seems to be an appropriate tool to fit the requirements for self-manageability of complex smart environments.

More research and development is needed to fully implement the opportunities, which GUN vision opens for the future generation smart spaces. Some of the ongoing efforts include recently started UBIWARE (“Smart Semantic Middleware for Ubiquitous Computing”) project (see: http://www.cs.jyu.fi/ai/OntoGroup/UBIWARE_details.htm), which aim is to deepen the SmartResource project results towards closer future of GUN.

References

[1] Berners-Lee, T., Hendler, J., and Lassila, O., The Semantic Web. Scientific American, 284(5), 2001, pp. 34–43.

[2] Buckley, J., 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, Brussels, 2006. Available in: http://www.rfidconsultation.eu/docs/ficheiros/WS_1_Final report_27_Mar.pdf .

[3] Helal A., King J., Zabadani H., and Kaddourah Y, The Gator Tech Smart House: An Assistive Environment for Successful Aging, Book Chapter in: H. Hagrass (Ed.), Advanced Intelligent Environments, Springer Verlag, 2008.

[4] 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), pp. 31-63.

[5] 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), pp. 55-78.

[6] 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.

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

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

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

[10] Nikitin S., Terziyan V., Pyotsia J., Data Integration Solution for Paper Industry - A Semantic Storing, Browsing and Annotation Mechanism for Online Fault Data, In: Proceedings of the 4th International Conference on Informatics in Control, Automation and Robotics (ICINCO), May 9-12, 2007, Angers, France, INSTICC Press, pp. 191-194.

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

[12] 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, June 12-16, 2007, Funchal, Madeira, Portugal, pp. 382-389.

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

[14] 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, Springer-Verlag, LNCS, 2889, pp. 279-291.

Acknowledgements

Author is grateful to all members of Industrial Ontologies Group to their contribution to SmartResource project, 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 JSP) for supporting activities of the SmartResource project. Thanks for Dmytro Zhovtobryukh for helping in finding motivating scenario and outlining benefits, which GUN approach can offer for the smart spaces domain. Author is also grateful to anonymous reviewers for constructive critics and suggestions how to improve the paper.

Authors

Vagan Terziyan has got his Engineer-Mathematician Degree in 1981 from Kharkov National University of Radioelectronics (Ukraine), then at the same university in 1985 he made Ph.D. degree and in 1993 Dr. Habil Tech. degree and got title of Professor in Software Engineering. Currently he is acting as Professor (Distributed Systems) in MIT Department, University of Jyvaskyla (Finland). He is leading the SmartResource and UBIWARE projects funded by Tekes and he is the Head of Industrial Ontologies Group. His research and teaching profile is design of distributed, intelligent Web applications, systems and services, which are: (a) able to automatically discover, compose and integrate heterogeneous components; (b) able to manage heterogeneous data sources; and (c) utilising for that emerging Knowledge-, Agent-, Machine-Learning-, Context-Aware- and Semantic Web- based technologies and tools.