Original idea of Semantic Web as next-generation of the Web assumes that besides existing content there is a conceptual layer of machine-understandable metadata, which makes the content available for processing by intelligent software, allows automatic resource integration and provides interoperability between heterogeneous systems. Initial orientation of semantic technology development to the Web digital resources resulted to omission from consideration of some other industrial domain resources such as various devices, processes and even humans. In this paper, the meaning of a "Semantic Web resource" is expanded to include also industrial objects (devices, machines, systems, etc) and humans (experts, maintenance workers, etc.) as resources and thus as a subject of semantic annotation. Elaboration of a specific adaptation mechanism for these types of resources from their natural environment to a Semantic Web environment is an important challenge for such expansion. Our intention is to make industrial devices (as well as other Semantic Web Resources) proactive in a sense that they can analyze their state independently from other systems and applications, initiate and control own maintenance proactively. In this research we join together Semantic Web, Web Services, Peer-to-Peer and Agent technologies into an integral resource management framework with resource-to-resource interaction aimed to improve maintenance of separate resources. The issues are also addressed related to implementation of gradually enhanced prototype of distributed Semantic Web enabled maintenance management environment. The environment assumes complex interactions of components, which are devices, humans (experts, operators) and remote diagnostic web-services. Future markets for such tools and resources have been pointed out.

PROACTIVE SELF-MAINTAINED RESOURCES IN SEMANTIC WEB

Olena Kaykova, Oleksandr Kononenko, Oleksiy Khriyenko, Vagan Terziyan, Andriy Zharko

Industrial Ontologies Group, MIT Department, University of Jyvaskyla, P.O. Box 35 (Agora), FIN-40014, Jyvaskyla, Finland

e-mail: **vagan@it.jyu.fi**

1.Introduction

The Semantic Web is an initiative of the World Wide Web Consortium (W3C) [1], with the goal of extending the current Web to facilitate Web automation, universally accessible content, and the 'Web of Trust' [2, 3]. Semantic Web is the vision of having data defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications [4]. The goal of Semantic Web development is promotion of existing Web to qualitatively new and higher level, utilizing machine-processable metadata associated with Web resources. Next generation of intelligent applications will be capable to make use of such resource descriptions and perform resource discovery and integration based on its semantics. Semantic Web approaches to development of global environment on top of Web with interoperable heterogeneous applications, web services, data repositories, humans, etc.

On the technology side, Web-oriented languages and technologies are being developed (e.g. RDF, RDF-Schema, DAML+OIL, OWL, DAML-S) [5-8], schema and ontology integration techniques are being examined and refined. The success of the Semantic Web will depend on a widespread adoption of these technologies.

Management of resources in Semantic Web is impossible without use of ontologies, which can be considered as high-level metadata about semantics of Web data and knowledge [9]. Ontologies are content theories about the sorts of objects, properties of objects, and relations between objects that are possible in a specified domain of knowledge. There is a growing interest in the use of ontologies in agent systems as a means to facilitate interoperability among diverse software components, in particular, where interoperability is achieved through the explicit modeling of the intended meaning of the concepts used in the interaction between diverse information sources, software components and/or service-providing software. The problems arising from the creation, maintenance, use and sharing of such semantic descriptions are perceived as critical to future commercial and non-commercial information networks, and are being highlighted by a number of recent large-scale initiatives to create open environments that support the interaction of many diverse systems (e.g. Agentcities [10], Grid computing [11], Semantic Web and Web Services). A common thread across these initiatives is the need to support the synergy between ontology and agent technology, and increasingly, the multi-agent systems and ontology research communities are seeking to work together to solve common problems.

There are several going on EU funded projects, which are targeting various aspects of emerging Semantic Web. Among most strong consortiums and initiatives are: OntoWeb [12] network with more than 100 academic and industrial participants, which creates a technical roadmap of the next generation Web and provides guidelines to industrial and commercial applications; SWAP [13] (Semantic Web and Peer-to-Peer), which provides a comprehensive study of the potential of Semantic Web and Peer-to-Peer for knowledge management and plan to provide an appropriate integrated software environment; SWWS [14] (Semantic Web Enabled Web Services), which is researching for scalable mediation between different and heterogeneous services based on semantic-driven descriptions and business logic; SEWASIE [15] (Semantic Web and Agents in Integrated Economies), which addresses the problem of access to heterogeneous data sources on the Web; SCULPTEUR [16] (Semantic and Content-Based Multimedia Exploitation for European Benefit), which is developing the technology to create, manipulate and manage cultural archives to make European cultural heritage accessible to all; MOSES [17] (Modular and Scalable Environment for the Semantic Web), which sets out to create scalable ontology based Knowledge Management System and ontology-based search engine that will accept queries and produce answers in natural language; and many other projects.

On the other hand, industry is looking for fast and global solutions related to Knowledge Management, Enterprise Application Integration, Electronic Commerce, Asset Management, etc. Various industrial standards, which have been created and implemented by different industrial consortiums, appear to be not sufficient for growing interoperability demands. Consider for example industrial maintenance management systems and solutions, which will be a subject of our pilot implementation. Many companies accumulate maintenance knowledge and develop own standards and systems for automated condition monitoring, diagnostics and maintenance and there is no easy way to enable sharing this knowledge among companies, automatic discovery of needed resource, connect diverse systems into next-generation interoperable industrial environment. However, such integrated environment can be created by utilizing Semantic Web standards and technology.

One of the recent initiatives aimed at development of adoption of open information standards for operations and maintenance and implementation of interoperable cooperative industrial environments is MIMOSA [18] (Machinery Information Management Open System Alliance). The project consortium pretends to build an open, industry-built, robust Enterprise Application Integration and condition-based maintenance specifications. There is also going-on large international project PROTEUS [19], funded by industrial companies and led with a goal to develop a generic maintenance-oriented platform for industry. These initiatives are very expensive, labor and resource consuming, and still does not attempt to apply and benefit from the Semantic Web technology. We believe, however, that without comprehensive metadata description framework, ontologies and open knowledge/semantics representation standards their results will be just next consortium-wide standards, rather than comprehensive, flexible and extensible framework.

At present, Web resources (web pages, web databases, web services, etc.) are meant to be consumed by humans only and have usually human-oriented representation. That is why resources cannot be easily processed by software meaningfully, i.e. taking into account semantics and relations with other resources. To be understood by software application, semantics must be presented explicitly in some form, which would allow intelligent information processing substituting human. Such semantic description is a metadata (data about data), attached to a resource. Addressing these problems W3C consortium has started Semantic Web Activity, which resulted in development of *Resource Description Framework* (RDF) as a basic model for semantic descriptions. In order to provide interoperability between heterogeneous software, semantic descriptions must be based on common terminology. Semantic Web assumes creation of vocabularies of concepts for specific domains and using these vocabularies (ontologies) for description of resources that will enable their automatic discovery and integration. Ontologies provide conceptual views of problem domains.

Currently, domains of Web content and Web services are in focus of Semantic Web Activity and semantic technology applications and correspondent ontologies develop most rapidly here. However, for industrial adoption of Semantic Web technology these efforts seem to be not enough. In our opinion, the problem is initial orientation of semantic technology development to World Wide Web digital resources. This resulted to omission from consideration of other industrial domain resources: devices, processes and even humans.

Resource discovery and integration (finding resources and use of semantically annotated data from many sources) are main concerns of Semantic Web applications, so appropriate support for semantic annotation is developed. At the same time, such aspects as description of *resource state* and *resource goals* are not considered. The latter are very important for such industrial applications as assets (resource) management, including resource condition monitoring, diagnostics and maintenance, which require use of such information in complex management activities.

In this paper we expand the meaning of the term "resource" in Semantic Web considering industrial objects (devices, machines, systems, etc) and humans (experts, maintenance workers, etc.) as resources and thus as a subject of semantic annotation and we focus on the problem of creating a global cooperative environment for automated industrial resource maintenance. The environment should enable automatic discovery, integration, condition monitoring, diagnostics, cooperation and learning of the heterogeneous resources for solving maintenance problems. Our intention is to combine innovative theoretical approach with practical applications and bring new values to appropriate businesses.

We base our research on recent ideas and results within Semantic Web technology applications in the field of industrial devices' maintenance. Among these there are concepts developed by Industrial Ontologies Group [20-26], such as:

- *OntoServ.Net, a Semantic Web based web-services* integration environment for industrial maintenance;
- *GUN – Global Understanding eNvironment*, a heterogeneous Semantic Web-based environment for agent-based resource management;
- *OntoAdapter* (semantic adapter/wrapper), a generic software component for connection of resources to semantic-enabled environments;
- *OntoShell*, a service platform for hierarchical service integration;
- *Mobile Resource*, i.e. technical approach for delivering agent-based services in distributed environments.

2.Global Understanding Environment

2.1.Proactive Resources in Semantic Web

Industry tends to build large-scale maintenance environments, where industrial instrumentation can be monitored, diagnosed and maintained remotely by automation systems [27]. Resources integrated in such environment are naturally heterogeneous at least in data presentation formats and in methods to access these resources.

Essential concern in resource maintenance is processing data about resource state. *State of the resource* can be understood in broader meaning than just values of some internal properties, but also as a relation between internal state (including its history), external factors and the purpose of resource existence. The analysis of the factors that influence state of the resource provides view to characteristics of balance between internals and externals of the resource, also meant as *resource condition*. The open standard for representation of states and conditions of complex industrial objects and processes is required for efficient resource diagnostics and maintenance by heterogeneous applications.

In the context of resource maintenance, the challenge is to create *Resource State/Condition Description Framework* (RSCDF), as an extension to RDF, which introduces upper-ontology for describing maintenance-oriented characteristics of resources: states and correspondent conditions, dynamics of state changes that happen, target condition of the resources and historical data about previous states.

Resources (e.g. devices) are assumed to have their own state presented as RSCDF descriptions. These descriptions are used by external applications (e.g. remote diagnostics) that support RSCDF and are able to process data presented in such format. Introduction of RSCDF allows solving problems of interoperability and resource heterogeneity (the same basic concepts will be used for state description of any kind of resources). Design of the RSCDF will follow the ontology engineering principles in the scope of Resource Description Framework developed by W3C Semantic Web Activity.

In Semantic Web, as presented by creators of this concept, resources are meant to be accessed, used and changed only by external applications or resource providers (owners). If there is some processing logic associated with a resource, then a responsible for its execution entity exists. Similarly, if to consider current maintenance systems with remote access, diagnostics and control, the resources are passive, waiting for intelligent tools to discover them and process their state.

There is growing interest in supplying (smart) industrial devices with condition monitoring, diagnostic and maintenance applications using integrated (embedded) computing systems that allow advanced data processing locally [28]. This improves and makes resource management more flexible, but such solutions still have specific software and its users (humans) as initiators and coordinators of maintenance process.

Our intention is to make devices active in a sense that they can analyze their state independently from other systems and applications, initiate and control own maintenance proactively. Resource state can provide knowledge about resource condition, whereas both resource condition and goal (purpose) of resource will result in certain behavior of active resource towards effective and predictive maintenance.

Resources, in our vision, will have integrated mechanism that allow flexible configuration of resource goals and behavior model. Behavior engine of resource includes support for detection of abnormal resource conditions via continuous monitoring, execution of appropriate behavior patterns striving for achievement of resource's maintenance goals. Implementation of such mechanism requires description of resource goals and models of resource's proactive behavior.

Resource Goal/Behavior Description Framework (RGBDF) is another extension of semantic resource description model RDF additionally to the resource state/condition description mechanism. Development of the models for resource state, condition, goals and behavior descriptions in the context of industrial maintenance is one of our main contributions to Semantic Web technology and future industrial applications.

In addition, we enable human presence in Semantic Web environment considering human to be a resource, not just a user in Semantic Web. Being naturally proactive, human can communicate with other resources and application acting as a web service. Removing the conceptual difference between human and web service and introduction of *human-resources* concept is another important challenge for development of automated resource maintenance systems.

2.2. GUN Concept

As a generalization of our approach, which enables resources as proactive components in resource-maintenance environments, we have developed the following concept.

Concept of Global Understanding eNvironment (GUN) is designated to join most recent research results, standardization effort, features and benefits of Semantic Web, Web Services, Peer-to-Peer and Agent technologies in integral resource management-oriented framework.

Basic features of GUN are:

- Semantic Web principles reused for integration of resources and interoperability;
- Proactive, goal-driven, self-maintaining resources;
- Wide spectrum of supported resource types;
- Resource-to-resource communication;
- Mobility of resources.

In GUN, resources are Semantic Web enabled in a sense that they are annotated using standards of ontology-based representation, in which semantics of separate piece of data identified by a reference to assigned meaning (a concept from ontology).

In GUN, resources are active participants of the environment (besides the applications that use them). This assumes adding associated resource-maintenance mechanism – resource-maintenance agent, which provides goal-oriented capabilities and support for self-interested, self-maintaining proactive behavior. Due to this, resources inherit agent features and become capable to have assigned goals. Mechanism of assigning goals and behavioral models to a resource can be considered as a higher-level development tool for resource maintenance applications.

Making a resource proactive and less dependent on continuous care from its provider can become a powerful approach to resource management in a dynamic distributed environment. Following it, resource can maintain itself in a way its provider specified depending on analysis of own state and conditions. In a very general way, the proactive resource can be defined as the one, which *has goal to maintain continuously balance between its internal and external environments*. It is assumed that *internal environment* of resource is its current state and external environment is everything the resource depends on or aware of, and which may influence the resource respectively to its goals.

Since agents are representatives of resources and realize their goal-driven behavior, the GUN is similar to a multi-agent system, in which resource integration is based on agent communication, coordination and negotiation. Agent's features of being a self-interested entity allow also embedding various business models into maintenance management environment.

Types of resources integrated into GUN are not limited only to digital documents and database content. Real-world objects can be also represented as resources capable, for example, to accept and respond to queries, interact with other resources in order to achieve own goals. Specific adaptation mechanism has to be elaborated for communication between resource and its agent. Taking into account great variety of possible resource types, diversity of their formats and ways of accessing them, adaptation of such resources in resource management environment will be an important challenge for development of GUN. General-purpose resource agents can be set up to maintain virtually any kind of supported resources using resource adapters. Development of resource adapters for each class of resources is required for GUN.

Another adaptation to be performed is an adaptation of resource semantics to environment where it resides. Initially none of the resources is semantically annotated and available for semantic-enabled environment. This is due to use of real-world resources and other types of resources that have to become Semantic Web enabled. In GUN adaptation is made by a resource agent, which "wraps" data retrieved from resource with semantic labels and delivers semantically annotated data from outside to a resource stripping out semantic markup.

Distinctive feature of the GUN comparatively to Semantic Web environment is resource-to-resource interaction with the aim to improve maintenance of separate resources via information sharing and use of other resources as web services.

Resource discovery in GUN utilizes semantic annotation of resources and appropriate peer-to-peer [29] semantic search models.

Mobility of GUN resources means that there is a support for models of preserving resource identity during changing resource-location site. Even transferred to a new location, resources continue their proactive behavior and interact with the environment. Necessity for mobility and its benefits can be shown on examples of resources performing actions that demand close, intensive and secure communication with other resources.

Additionally to original challenges provided by the Semantic Web approach:

- *resource discovery* based on semantic descriptions*;*
- *semantics-based resources integration;*

the GUN concept assumes at least two new challenges:

- *resource state monitoring:* retrieval, semantic annotation and accumulation of a resource state data and its initial preprocessing for discovery of special classes of states (alarms) that require advanced handling and initiate resource-maintenance mechanisms;
- *resource state diagnostics/maintenance:* learning models of resource state classification (using various machine learning techniques), automated testing, quality evaluation and certification of learned models and application of models for diagnostics and maintenance decision making.

In the context of the industrial maintenance case, resources are considered as complex industrial objects [30] with dynamic internal state, which need continuous condition monitoring and which have embedded processing capacity to proactively realize self-interested maintenance-oriented behavior and communicate with external systems in order to achieve their own goals and improve maintenance performance.

Thus, we are implementing the GUN concept as a *cooperative environment for predictive maintenance* build on top of existing industrial infrastructure using Semantic Web, Agent Technology, Peer-to-Peer and Web Service technologies for development of an integrated solution.

3.Self-Maintained Resources

3.1. Features of Self-Maintenance Environment

We want to advance the vision of self-maintained resources in a "Global Understanding Environment" to the extent, which allows developing basic technology and software tools. The industrial maintenance case has been chosen for implementation of the pilot system and testing the research results obtained. In given case study, resources are industrial machines that require monitoring and maintenance. Other supported resources will be experts in diagnostics (human and intelligent software), diagnostic web services (existing applications adapted to web), and maintenance centers (decision tools or other applications that support industrial maintenance).

Our system supports the following main tasks and features:

1. Industrial object (resource) adaptation to the Semantic Web environment;
2. Condition monitoring of the resource (by an external system or by the resource itself) enabling semantics in resource state descriptions;
3. Alarms detection (performed by external system or the resource itself). Alarm is considered as condition that requires more detailed examination and diagnostics;
4. Diagnostics of the resource by remote diagnostic services;
5. Support for complex models of resource-maintenance and their automated execution;
6. Integration of heterogeneous diagnostic/maintenance services, enabling resource-to-resource communication and peer-to-peer service/resource discovery;

Research concerning self-interested goal-oriented behavior of resources will be done in the context of self-maintained industrial resources (devices). Resource Goal/Behavior Description Framework (RGBDF) will be designed as background for future implementation of generic mechanism that allows resource itself "understand" changes of its state and perform self-maintenance. Such *behavior engine* can be used as integral part of proactive resources in industrial maintenance-oriented environments (condition-monitoring and maintenance systems). RGBDF will include ontology, describing resource goals concepts and relations between them, formal language for programming resource behavior, basing on mentioned ontology, and module, which interprets program of resource behavior.

Software development goals of the paper include implementation of gradually enhanced prototype of distributed maintenance environment with complex interactions of components, which are devices, humans (experts, operators) and remote diagnostic web-services. Maintenance agents are added to industrial devices, which are interconnected in decentralized network and can integrate diagnostic services in order to increase the quality of maintenance for individual devices.

Final development goal will be achieved via accomplishing the following tasks:

1. Design of Resource State/Condition Description Framework (RSCDF) and implementation of resource browsing mechanism, which allows external entity (intelligent resource maintenance tool, service or human-user) to follow the changes of resource state via device-independent Semantic Web-based data-exchange protocols and syntaxes. RSCDF will include ontology, describing resource states and conditions and design of resource maintenance tool;
2. Development semantic adapters for devices, maintenance services and humans and improving human interaction with Semantic Web-based environment representing human also as a resource or a web service that can be invoked/queried by other resources. Human will interact with the environment via special communication and semantic adapter;
3. Design of agent-based resource maintenance models in distributed environment;
4. Development of P2P infrastructure and implementation of the prototype components of Semantic Web enabled maintenance environment.

Using maintenance case, we are going to demonstrate the benefits and newly opened possibilities of developed resource management framework and promote newest information technologies to Finnish industry.

3.2. Challenges of Self-Maintenance Environment

Development of GUN environment for achieving paper goals will meet the following challenges:

3.2.1. Resources are proactive and perform self-maintenance via proactive behavior

Proactive behavior of self-maintained resource requires that in the implemented system:

- resource should try (first itself) to figure out its own maintenance needs;
- resource should try (first itself) to perform its own maintenance with locally available tools;
- resource should be able to communicate with other similar resources, to ask them for assistance, forward the problem to the one, which is more experienced, get diagnosis and understand it;
- resource should be able to update maintenance models, to ask for support of remote maintenance services and required tools from other resources/services or a human wrapped as external services.

In addition, a resource should be able to predict its current state in dynamics, and what is more important, its new semantics for predictive maintenance when necessary. Such behavior of resource can be applied in condition monitoring, remote diagnostics and maintenance domain [31].

Resource State/Condition Description Framework both with Resource Goal/Behavior Description Framework provide a basis for development of interoperable representation for generic software components used for creation of resource-maintenance agents.

Implemented prototype of maintenance system should include a maintenance agent integrated into resource (device) with requirements that:

- maintenance agent analyzes device state by applying set of locally available tools for preliminary diagnostics;
- agent's behavior is specified in a goal-descriptor, which contains interpretable "program" of actions: events it should support, prerequisites for performing actions, action parameters and other required data for that;
- maintenance agent requests additional tools and services for device maintenance from network, which are relevant to current maintenance activity;
- maintenance agent notifies external device-maintaining resources (other automatic condition monitoring, maintenance system and also humans) and forwards the problem if it detects situation, which requires that.

3.2.2. Industrial machines, human-experts and diagnostic services are semantic-enabled resources

GUN supports resources of any possible type virtually. Development of resource adapter architecture and specific resource-adapters allows inclusion of humans, devices and other non-traditional resources into GUN.

In the maintenance case implementation, devices, maintenance software systems and services, and humans (device-supporting personnel) are considered as maintenance system components.

Firstly, adaptation is required to specific way of communication with new types of resources. Secondly, *semantic adaptation* of resources is applied in order to make them available (retrieval of their data, state or description is possible) for other resources in the heterogeneous environment.

Development of resource-specific semantic adapters is considered as an important step towards semantic-based communication in heterogeneous environments, in which interoperability problem is solved using common ontology and representation-neutral syntaxes for data exchange.

3.2.3. Resources can learn models of own annotations

One of important challenges in our paper is resource-maintenance learning. Resource maintenance agent in implemented prototype should be able to learn (either via supervised or unsupervised learning techniques) to try to annotate semantically new version the resource without external support when possible.

Considering maintenance case, if a resource has samples of labeled (already interpreted and annotated) data about its states (e.g. history of previous states, exceptional cases, failure states, etc.), then learning tools try to build model from this data and use it for interpretation of new states, labeling own state and publishing its annotated in a resource description.

In addition, resource learning is enhanced with possibility to utilize other than local learning models, discover, exchange and integrate them within a GUN network, if the local model is not enough. Sharing of learned (maintenance) models in GUN promotes better performance of resources. Research efforts are required for exploration of existing methods and selection for implementation of the most appropriate types of learning techniques.

3.2.4. Resources are organized in resource-to-resource (R2R) communication network

Peer-to-peer communication (without centralized control) is a natural way to communicate in heterogeneous environment. Nevertheless, development of communication framework for GUN environment will consider both centralized and peer-to-peer models.

Resources can use search mechanisms provided by GUN, built on top of underlying communication framework. Resources can search for other resources, solutions or tools for problem provided as query to GUN.

Implementation of peer-to-peer capabilities requires additional research efforts comparatively to centralized search. Most challenging factor is that semantic-based search techniques are to be developed.

Implementation of maintenance model applies Semantic Web technologies (ontology formalism, ontology languages, Resource Description Framework) in order to provide interoperability between maintenance systems components, possibility to discover resources, exchange (and integrate) the learned models. Maintenance domain ontology is required for that in order to fix semantics of terms used in communication, thus, supporting communication at the level of semantic concepts.

4. Implementation Aspects

For the implementation we are using the most emerging technologies related to Semantic Web, Ontology Engineering, Semantic Annotations, Semantic Search, Intelligent Web Services (Modeling, Discovery and Integration), Multi-agent Systems, Mobile Agents, Peer-to-Peer Networks, etc.; standards from W3C and FIPA [32], various Artificial Intelligence methods. System analysis and object oriented design and programming with UML, Java and XML is applied for the prototype implementation. Ontology engineering is performed using W3C standards and DAML-OIL (OWL) in Protege 2000 [33] environment.

Software development stages of the project include implementation of gradually enhanced prototype of distributed maintenance environment with complex interactions of components, which are devices, humans (experts, operators) and remote diagnostic web-services.

4.1. A Generic Resource Access Mechanism (Semantic Adapter)

4.1.1. Model components:

- ***Semantic Adapter***, generic software component for connecting resources;
- ***Device-Data Source***, an object that is either device accessed directly via specific hardware/software interface or another kind of device state-data source: database or emulated (virtual) device;
- ***Resource Browser***, a software that provides access to resources with semantic adapter via semantic interface; browser has user interface (UI) and can be used as part of semantic adapter to Human;
- ***Human***, a person involved into activities performed in the system;
- ***Service Component***, a standalone software component (executable, program code, dynamic library) or web service used for performing some servicing actions.

Database managed by FieldBrowser™ [34] can be one of the possible sources of device-data.

4.1.2. Component requirements:

- Semantic Adapter (see Figure 1) is used with resources of every type (Device-Data Source, Human, Service Component) in order to:
  - make resource's data available via semantic queries;
  - execute semantic queries on other resources;
- Device-Data Source and Service Components are passive, all operations on them are initiated by other resources and executed by attached semantic adapter;
- Human uses Resource Browser for acquiring data from resources; Resource Browser uses queries to resources expressed in terms of Messaging Ontology (for general structure of query-messages) and Maintenance Ontology (for specifying concrete data values and resource-specific commands);

- Service components expose some program interface, either API or SOAP-interface [35] (for web services)

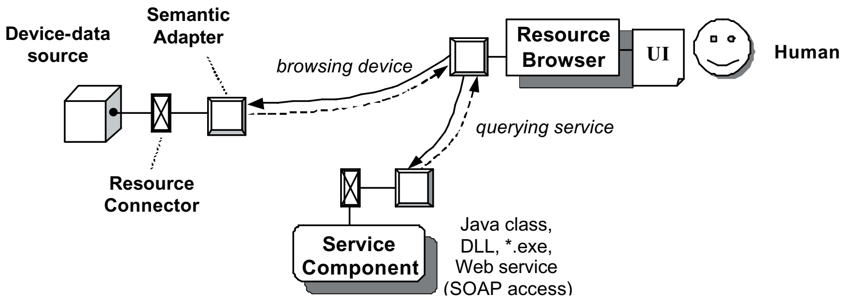


Figure 1. Semantic Adapter and related components

4.1.3. Software components:

- Simple service components implementation – used for demonstration of system operation;
- *Generic semantic adapter* for devices, services and humans, that translates semantic messages into native actions of adapted object (device, human, diagnostic algorithm) and vice versa; translation is configured for every object separately;
- Specific connectors used with semantic adapters for:
  - device-data (for access to database of FieldBrowser™, or other, via standard semantic interface)
  - human (Resource Browser is reused for this purpose)
  - service (for standard programmatic access to software components/web-based services)
- *Network Connector* – reusable software component that supports common (as implemented in current version) transport-level connections between ***Device(s), Expert(s)*** and ***Service(s)***.

4.1.4. Supporting ontologies:

Messaging Ontology: support for data exchange protocol between ***Semantic Adapters*** – introduction of general terms for construction of semantic-based messages;

Maintenance Ontology:

- support for device state description (annotation of device parameter values)
- support for service description (descriptions of data required for learning and diagnostics)
- support for history representation (general structures for device history description)

4.2. Simple Remote Diagnostics Model

4.2.1. Model components:

- ***Device***, a sample of a device, which state is to be automatically annotated with "diagnosis" (a device state classification label); supplied with *Local Alarm Service* linked via standardized semantic interface. *Local Alarm Service* is a local device-specific algorithm capable to detect alarm states of the ***Device***.
- ***Expert***, a source of classification results for given device state that are assumed to be correct (human);
- ***Service***, a standalone diagnostic algorithm capable to learn classification models and present to device possibility to use it as a tool for self-diagnostics.

Database managed by FieldBrowser™ can be one of the possible sources of device-data. ***Device, Expert*** and ***Service*** are semantic-enabled ***Device-Data Source, Human*** and ***Service Component*** (each component has a ***Semantic Adapter*** and supports common communication mechanisms). Maintenance-oriented behavior is realized by specific ***Interaction Module***.

4.2.2. Component requirements:

Device:

- Collects and accumulates own history data, store device state data with classification label, if available;
- Supports monitoring made by *Local Alarm Service;*
- Supports monitoring made by ***Expert:***
  - Accept queries using semantic descriptions of requested data
  - Provide requested data
  - Accept labels (if returned) for device states and store them in history
- Uses ***Expert*** and ***Service*** for (automatic) labeling state of ***Device:***
  - Get semantic description of data required for classification
  - Provide required data
  - Get classification result (device-state label) and put it to the history
- Supports learning of ***Service*** (batch learning):
  - Accept semantic description of requested by ***Service*** data (for learning)
  - Supply all requested data

Expert:

- Accepts semantic description of device state and can respond with classification label (semantic description of classification);
- Can request with semantic query device-state data (also history and made by ***Service*** labels) for watching, get response from ***Device*** and provide own label for observed device state.

Service:

- Accepts semantic description of device state from ***Device*** and respond with classification label obtained using existing learned model for classification;
- If classification model has to be built first (no model yet) than perform learning:
  - Request data required for learning using semantic query
  - Build (via learning technique) classification model
  - Notify ***Device*** about readiness to perform diagnostics

4.2.3. Diagnostic model structure

General structure of the diagnostic model is presented in Figure 2.

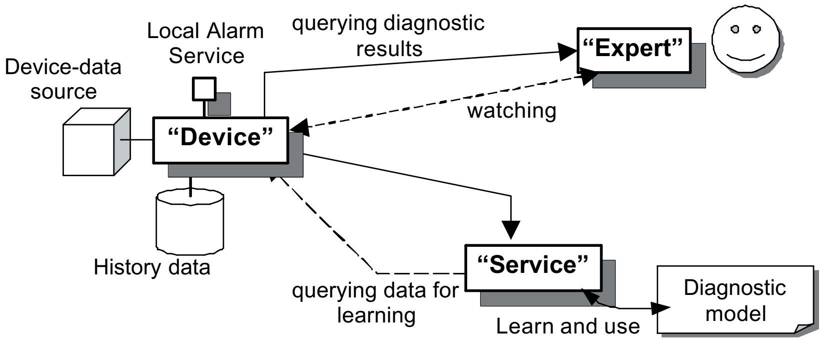


Figure 2. Remote diagnostics model and related components

4.2.4. Software components:

- Simple service implementation – supports behavior of ***Service*** as described;
- *Local Alarm Service* will be implemented as simulator of real industrial alarm system or will represent adopted and adapted real industrial alarm system;
- *Interaction Module* – software component that realizes resource behavior.

4.2.5. Supporting ontologies:

- *Messaging Ontology* is extended in order to support simple networking;
- *Maintenance Ontology:* extended support for querying data from Device.

4.3. Peer-to-Peer Maintenance Management Environment

4.3.1. Model components:

- Set of ***Devices*** (device agents), representatives of monitored devices, which states are to be automatically annotated with "diagnoses", a device state classification labels provided by diagnostic services and human-experts; supplied with *Local Alarm Service;*
- Set of ***Experts*** (human-agents or trusted diagnostic service), services represented by humans, which can give classification results for given device state that are meant to be correct;
- Set of ***Services*** (service-agents), standalone diagnostic services capable to learn classification model of expert, allow devices use it as a source of own state diagnoses;
- Set of ***Notice-Board*** services that store advertisements of ***Devices, Experts*** and ***Services*** and support publishing advertisements and filtering board contents with semantic query.

Devices, Experts and ***Services*** have own contact lists with links to other network components, which they can interact with for own purposes: exchange of data, querying services, etc. These links form *peer-to-peer network* [36], in which any peer using semantic query can perform search for other components.

In this version ***Device*** will support service composition in form of ensembles using own models of service quality estimation. Service composition is made with goal of increasing diagnostic performance

Semantic query is a special kind of search expression presented using semantically annotated data; it can be interpreted correctly by queried party; it is assumed that common ontology of such queries exists and is supported by all peers.

Notice-Boards are entry points into peer-to-peer network and play important role in creation of peer-to-peer environment.

4.3.2. Component requirements

Device: (now supports also an integration of multiple services)

- Collects and accumulates own history data, stores device state data with classification labels, if available:
  - uses diagnostics from several reference ***Experts***
  - uses provided labels for correction of ***Services'*** quality estimate, start using ***Experts*** as classification source, if quality of ***Services*** gets too low
  - uses some models for obtaining generalized diagnostic results from many ***Services*** along with separate results, which will be used for learning quality of ***Services***
- Supports monitoring made by *Local Alarm Service*;
- Supports monitoring and control over device behavior (configuration) performed by ***Experts***;
- Requests diagnostics from several ***Experts*** and ***Services***, in order to increase diagnostics quality via combination of obtained results, for (automatic) labeling state of ***Device***;
- Supports learning needs of ***Services*** and can manage it (selects sample data for learning):
  - both online and batch learning techniques are supported (with appropriate protocols)
  - generalized results of device state classification from device history can be used (generalized opinion of ***Experts***, other ***Service***, or both combined);
- Tests ***Services*** in order to estimate "quality" of ***Services***:
  - using some of already labeled by ***Experts*** states from history
  - using ***Device*** data and opinions of other ***Services***
- Uses ***Services*** instead of ***Experts*** if its quality estimation is higher than certain threshold level;
- Searches for new ***Services*** and ***Experts*** with goal of increasing diagnostic quality:
  - using ***Notice-Board*** services
  - peer-to-peer search via neighbor contacts (direct links to other network components)
- Advertises itself to other components in existing P2P network and via ***Notice-Board*** services;

- Exchanges learned by own estimates of ***Services*** and links to them with other ***Devices***.

Expert (now supports servicing multiple devices):

- Accepts diagnostic requests from multiple ***Devices*** and can respond with diagnostic result (which is semantically annotated description);
- Can request with semantic query device-state data (also history and made by ***Service*** labels), get response from ***Device*** and provide own label for observed device state;
- Can perform searches for new ***Device***-clients in existing P2P environment specifying semantic query;
- Advertises own presence to other network components via ***Notice-Board*** services;

Service (now supports servicing of multiple devices and learning from multiple sources):

- Accept semantic description of device state from ***Device*** and respond with classification label obtained using existing learned model for classification;
- Build classification model; many techniques are possible, e.g.
  - one model from several devices of same type;
  - own model for every device;
  - some combination of them, etc.
- Searches for new ***Device***-clients via neighbor's (e.g. other ***Services***) or ***Notice-Board*** services;
- Advertises itself to other components in existing P2P network and via ***Notice-Board*** services;
- Exchange ***Device*** addresses with other ***Services***;

Notice-Board service:

- Allows publishing advertisements of the network components;
- Filters published advertisements basing on income semantic query form network components and returns links to objects with matching advertisement;

4.3.3. Software components:

- ***Notice-Board*** services;
- Mechanism of advertisement;
- Search mechanism;
- Mechanism of service composition [37];
- Mechanism of service ensembles learning.

4.4.4. Global diagnostics model structure

Extension of maintenance model is proposed for further implementation with certification mechanism, which assumes that besides local devices' estimates of service quality, some certification authority exists that can provide certificates for service, which, in turn, will be used by devices for optimal service search and selection. ***Device*** also will be able to use estimates of service quality of external resources for building trust networks.

Condition-monitoring models can be extended by maintenance services, which can influence device state and maintenance actions on it. This also requires enhancement of device connectors, proper support of device adapter configuration, and introduction of new "device-maintenance center-operator" interaction models. These will finish the minimal working set of maintenance system components.

The structure of the global diagnostics model based on peer-to-peer network of interacting components is shown in Figure 3.

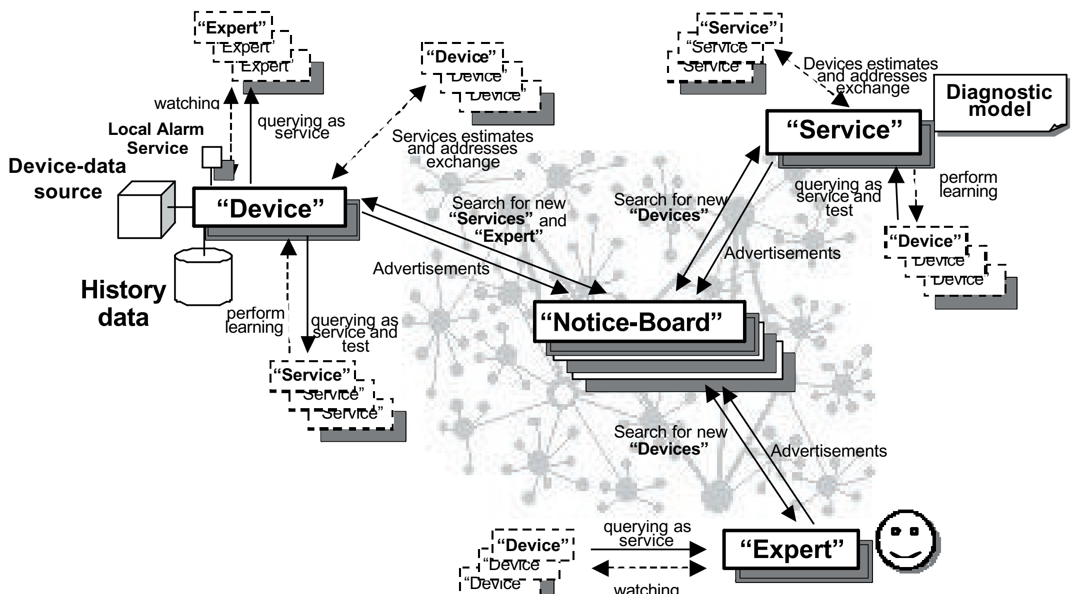


Figure 3. Global diagnostics model and related components

5. Overview of Future Markets for Products based on GUN Technology

The idea of proactive resources interconnected into global Semantic Web-enabled network (Global Understanding eNvironment - GUN) is aimed to be put to a future market of software products and services.

5.1. Products

Development of the GUN technology provides several main results, which can be lately commercialized:

- *Generic semantic wrappers for semantic adaptation of resources.*

Semantic adapter unifies communication with different types of resources (software, database, web content, human, industrial device, house appliances, OS, etc.) irrespectively of their great diversity and allows presenting them as a part of the semantic-based environment.

Development of the Semantic Web-based systems with non-digital (not a web content, physical resources – devices, humans that are available as resources) or not annotated yet resources requires efforts to "wrap" them with mechanism that will hide their non-digital and non-semantic nature and will provide support for semantic access to and semantic communication among resources.

Generic mechanism for that will be valuable both for software development and enabling new types of users and resources of the Semantic Web (devices, human, etc.) SmartResource project includes design and implementation of semantic adapters for such completely different kinds of resources as a software system (web service), a human (user of a semantic environment), an industrial device (that is to be maintained via communication with other resources).

▪ *Agent-shells for proactive resources*

This kind of software is aimed to provide "behavior" of resources (ability to perform autonomously actions in order to achieve some "goals" of resources' existence). Resources in this context are objects that have some internal state, which needs continuous maintenance accordingly to the rules specific to this resource. Resource "environment" (a part of the resource's world beyond itself) also influences the state of the resource and its behavior. Agent-like features of the resource can be enabled by using configurable shell, which utilizes descriptions of its desired behavior. Agent shell also solves problems of interoperability between object and supports communication with other resources "covered" by similar shell. Applications of the Semantic Web technology in the agent-shell software is necessary, as well as use of semantic adapters for resources.

By providing generic technology and tools for development of agent extensions to the existing resources, various resource management frameworks can be designed and implemented. Some examples for that are:

– Document management systems with documents that annotate themselves and update their contents from other documents and external sources;

– Distributed information systems, where autonomous data objects communicate in order to share information and perform their duties, etc.;

– Self-maintained devices that search and use appropriate maintenance services when needed.

These and other application cases can benefit from use of the generic agent-shell technology.

▪ *Technology and toolkit for building semantic-enabled environments*

Developed technology supposed to find many applications in various domains, in which distributed heterogeneous resources exist and problems of interoperability and integration into dynamic open environments are emerging.

The ultimate goal of the activities it this direction is a design of an innovative by its features environment (based on Semantic Web vision) and software design and development frameworks for the future generation of global communication system.

Parts of the GUN-technology include:

– Specification of the development process of the Semantic Web-based system;

– Specification of the ontology support for system components, which includes design of information management models and development of basic ontologies for needs of the system;

– Specification of the resource adaptation framework (semantic adapter development technology);

– Specification of the resource-to-resource communication framework that defines possible architectures of the communication subsystem, messaging protocols, search mechanisms, support for semantic-enabled components, etc.

Among supportive tools for design, development and deployment of GUN-based systems are ontology development tools, implemented semantic adapters and agent-shells with and configuration tools for them, browsing tool for semantic-enabled resources in the system that can be reused in the development of user interfaces for semantic-enabled environment.

Results of the *SmartResource* project will bring expertise how to apply GUN concept ideas for industrial maintenance domain, and other potential applications will be outlined for future implementation.

5.2. Customers

The classification of the potential customers that are in the market for the GUN-based technologies and products is presented in the following categories:

Providers of integration services

Such customers provide the Web platforms, portals, hubs, etc., which enable integration of heterogeneous resources for various purposes. They supply their own customers with a package that implements all necessary functionality of the target system (resource adaptation, maintenance agent, search, etc.). Service providers can also extend their platform by new categories of supported resources by implementation of new adapters or buying them from correspondent company-developer. For example, industry solution providers will interconnect heterogeneous industrial devices and services for them into a global (or enterprise-wide) maintenance system.

Software developers

Developers of software products (web applications, services, agents, etc.), who want to develop new products for integration environments supplied to end-users by the providers of integration services. They use GUN-technology and supportive toolkit for development of new applications as well as for adaptation of existing ones. Development principles of the GUN-based system (which is a Semantic Web-based environment with proactive resources) will be applicable for implementation of many semantic-enabled software products. Technology of semantic adapters and agent shells can be also reused for development of the products that require solving interoperability and integration problems, deployment of resource-to-resource communication systems or can reuse agent shells for own purposes.

Service consumers

Service consumers are end-users of the information and resource integration systems developed based on the GUN-technology. Depending on the kind of the system, the users could be individuals (e.g. users of Semantic Web-based file sharing network) and companies, i.e. those who want to get whole system for own purposes (e.g. company needs document management built on Semantic Web principles). See profiles of possible consumers and their interactions in Figure 4.

It is required for service consumers to use software products that make them part of the GUN-based environment.

They use various browsing tools, specific system applications and available services. Also they may need specific semantic adapters in order be connected to the system.

See general schema of future market of products based on GUN technology in Figure 5.

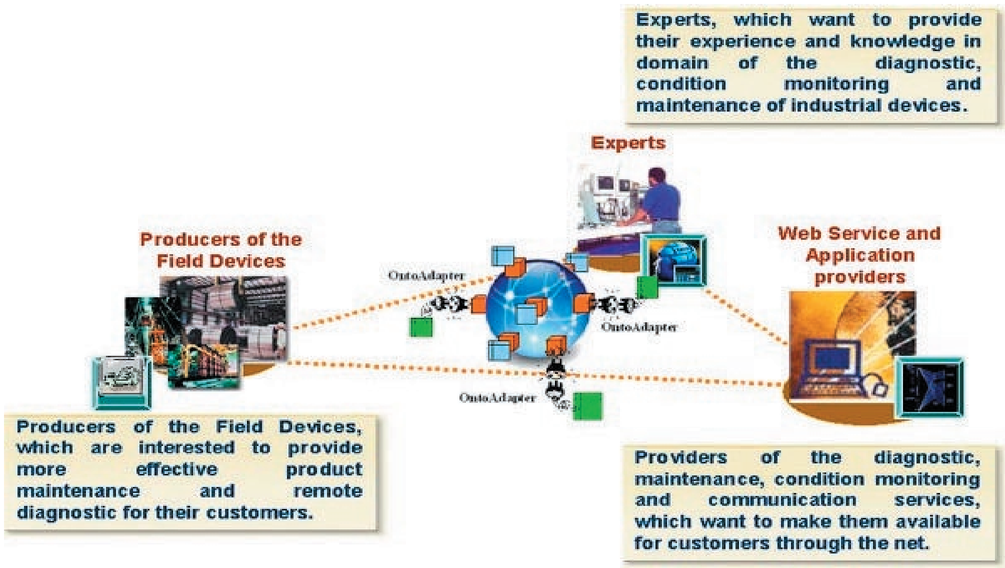


Figure 4. Customers of the maintenance system for industrial devices

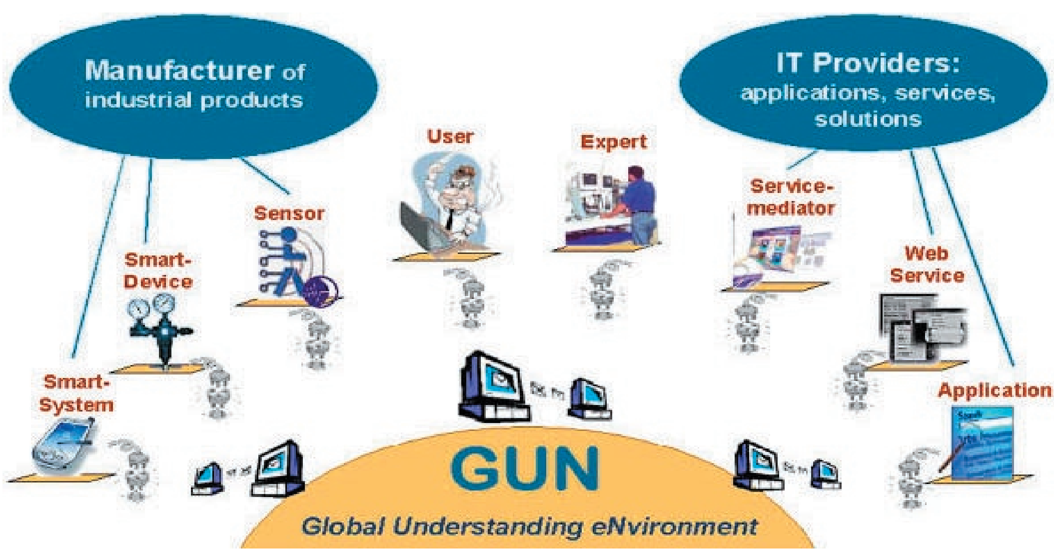


Figure 5. Market of the GUN-based products

6. Conclusions

The idea of proactive self-maintained resources in Semantic Web described in this paper corresponds to the emerging demands of the European enterprises. The initial application domain is maintenance of industrial instrumentation, but other domains (e.g. document management systems, distributed information systems) could benefit from it as well. The proposed solution is based on the concept of Global Understanding Environment, which utilizes the potential of synergy of the latest information technologies (Semantic Web, Intelligent Web Services, Multi-agent Systems, Peer-to-Peer Networks, Artificial Intelligence methods) and thus requires a comprehensive research for its further development. The research challenges and requirements to the system components defined in the paper are milestones for further analysis and the detailed elaboration of the outlined implementation strategy will facilitate the deeper analysis of the system. For instance, the development of Semantic Adapters will require overview of the existing software interfaces related to digital industrial resources. Development of the Maintenance ontology will require review of the industrial standards and classification of the industrial objects (e.g. available nowadays different types of industrial services). Peer-to-Peer Semantic Web based search is a huge task studied by many researchers.

The preliminary analysis of the market for this system has brought attractive results due to the universality of the resources being integrated into the GUN-environment. Three types of the products based on the proposed system (Generic semantic wrappers for semantic adaptation of resources, Agent-shells for proactive resources, Technology and toolkit for building semantic-enabled environments) and tree types of the potential customers (Providers of integration services, Software developers, Service consumers) that have been defined support the advisability of further development of the proposed ideas.

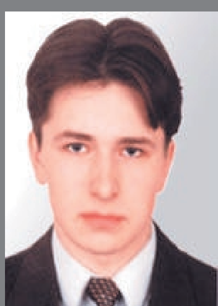
Vagan Terziyan 1958, received his Engineer-mathematician degree on Applied Mathematics from Kharkov National University of Radioelectronics (KhNURE) in 1981. He became Candidate of Technical Sciences (Dr. Tech. equivalent) in 1985 and Doctor of Technical Sciences in 1993 (Dr. Habil Tech. equivalent) at the Software Engineering Department. He is acting as Professor on Software Engineering since 1994 and as the Head of the Department of Artificial Intelligence and Information Systems since 1997 in KNURE. Area of research interests and teaching includes but not limited by the following: Intelligent Web Applications, Distributed AI, Agents, Multiagent Systems and Agent-Oriented Software Engineering, Semantic Web and Web Services, Peer-to-Peer, Knowledge Management, Knowledge Discovery and Machine Learning, Mobile Electronic Commerce. Recently he is working as Associate Professor in MIT Department, University of Jyvaskyla and as Senior Researcher at the InBCT (Innovations in Business, Communication and Technology) TEKES Project in Agora Centre and Head of "Industrial Ontologies Group". He has more than 100 scientific publications, more than half of them in internationally recognised magazines and conferences. He supervised about 10 research and software development projects.

Helen Kaykova 1960, received her Engineer-system-analytic degree on Automatic Control Systems in 1982 from Kharkov State Technical University of Radioelectronics (KhNURE). She became Candidate of Technical Sciences in 1989 on Technical Cybernetics and Information Theory. She was acting as Docent of Software Department since 1992, then as Docent of the Department of Artificial Intelligence and Information Systems since 1997, and then as Docent of the Artificial Intelligence Department since 1999. She was acting as the Deputy Vice-Rector on International Research Co-operation since 1996 and since 1999 as a Co-ordinator of International and European Affairs in KhNURE. Area of research interests includes: Artificial Intelligence: Logic, Temporal Reasoning, AI and Statistics. She is the Director of the Metaintelligence Laboratory in KhNURE, the Co-ordinator of co-operation and exchange Program with University of Jyvaskyla (Finland). She is now acting as Study Advisor on Mobile Computing in M.Sc. Program in the University of Jyvaskyla since October 2000. Since January 2003 Helen is a member of the "Industrial Ontologies Group" research group. She has near 40 scientific publications, 10 in internationally recognised magazines and conferences.

References

1. Official website of the World Wide Web Consortium (W3C), http://www.w3c.org/, last accessed 11th Dec 2003.
2. Web page of the Semantic Web Activity of W3C. http://www.w3.org/2001/sw/, last accessed 11th Dec 2003.
3. Golbeck J., Parsia B., Hendler J. Trust Networks on the Semantic Web. Proceedings of Cooperative Intelligent Agents 2003, August 27-29, Helsinki, Finland.
4. Berners-Lee T., Hendler J., Lassila O., The semantic web. Scientific American, 284(5), 2001, pp. 34–43.
5. W3C Candidate Recommendation, "Resource Description Framework (RDF), Schema Specification, 1.0", 27 March 2000, http://www.w3.org/TR/2000/CR-rdf-schema-20000327/.
6. Joint US/EU ad hoc Agent Markup Language Committee. Reference description of the DAML+OIL (March 2001) ontology markup language. http://www.daml.org/2001/03/reference, March 2001.
7. The DAML Services Coalition, "DAML-S: Semantic Markup for Web Services", May 2003, http://www.daml.org/services/daml-s/0.9/daml-s.html.
8. Dean M., Connolly D., Harmelen F., Hendler J., Horrocks I., McGuinness D. L., Patel-Schneider P. F., and Stein L. A. "OWL Web Ontology Language 1.0", Reference. http://www.w3.org/TR/owl-ref/, July 2002.
9. Mizoguchi R., Kozaki K., Sano T., Kitamura Y. "Construction and Deployment of a Plant Ontology". EKAW 2000: 113-128.
10. Official website of the AgenCities project, http://www.agentcities.org/, last accessed 11th Dec 2003.
11. Official website of the Grid Computing Information Centre, http://www.gridcomputing.com/, last accessed 11th Dec 2003.
12. Official website of the OntoWeb community, http://www.ontoweb.org/, last accessed 11th Dec 2003.
13. Official website of the SWAP (Semantic Web and Peer-to-Peer) project, http://swap.semanticweb.org/, last accessed 11th Dec 2003.
14. Official website of the SWWS (Semantic Web Enabled Web Services) project, http://swws.semanticweb.org/, last accessed 11th Dec 2003.
15. Official website of the SEWASIE (Semantic Webs and Agents in Integrated Economies) project, http://www.sewasie.org/, last accessed 11th Dec 2003.
16. Official website of the SCULPTEUR (Semantic and Content-based Multimedia Exploitation for European Benefit) project, http://www.sculpteurweb.org/, last accessed 11th Dec 2003.
17. Official website of the MOSES (Modular and Scalable Environment for the Semantic Web) project, http://www.hum.ku.dk/moses/, last accessed 11th Dec 2003.
18. Official website of the MIMOSA (Machinery Information Management Open System Alliance) association, http://www.mimosa.org/, last accessed 11th Dec 2003.
19. Official website of the PROTEUS project, http://www.proteus-iteaproject.com/, last accessed 11th Dec 2003.
20. Terziyan V., Semantic Web Services for Smart Devices Based on Mobile Agents, In: Forth International ICSC Symposium on Engineering Intelligent Systems (EIS-2004),

Oleksandr Kononenko 1981, graduated from the Kharkov National University of Radielectronics in June 2003 with Diploma of Engineer in Intelligent Decision Support Systems (with distinction). During 2003 he successfully graduated from the University of Jyvaskyla, Finland, Dept. of Mathematical Information Technology, Mobile Computing study line, and obtained Master of Science degree. As a member of the "Industrial Ontologies Group" research group he participates in the Semantic Web-related studies as a researcher/developer at the Agora Research Center, University of Jyvaskyla. His research concerns Semantic Adaptation of Resources in the Semantic Web-based environments.



Oleksiy Khriyenko 1981, obtained his Engineer's degree in Computer Sciences (a study line is Intelligent Decision Support Systems) in June 2003 from the Kharkov National University of Radioelectronics (KhNURE) in Ukraine. Also he is Master of Science, Dept. of Mathematical Information Technology (a study line is Mobile Computing). He obtained this degree in December 2003 from University of Jyvaskyla in Finland. He is also a member of the "Industrial Ontologies Group" research group since January 2003 and researcher in Agora Center (research center in Jyvaskyla, Finland). His research interests include: Artificial Intelligence, Semantic Web, Agent Technology, Web-Services, Distributed Resource Integration, and the industrial application of these and new technologies. (http://www.cc.jyu.fi/~olkhriye).

Andriy Zharko 1981, graduated from the Kharkov National University of Radielectronics in June 2003 with Diploma of Engineer in Intelligent Decision Support Systems. In December 2003 he successfully finished the Master's program at the University of Jyvaskyla, Finland, Dept. of Mathematical Information Technology, where he obtained a degree of Master of Science in Mobile Computing study line. He is also a member of the "Industrial Ontologies Group" research group since January 2003 and a researcher/developer at the Agora Research Center, University of Jyvaskyla. His research is concerned with Peer-to-Peer Semantic Web based large-scale systems.

Island of Madeira, Portugal, February 29 – March 2, 2004 (to appear).

21. 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.
22. 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, LNCS, Vol. 2853, Springer-Verlag, 2003, pp. 183-197.
23. Terziyan V., Intelligent "Mirror Web Browsing" vs. Pull/ Push Technology, Eastern-European Journal of Enterprise Technologies, V. 1 No. 1, 2003, "Technology Center" Foundation, ISSN: 1729-3774, Kharkov, Ukraine, pp.4-14.
24. Kaykova O., Kononenko O., Terziyan V., Zharko A., Formation Scenarios in OntoServ.Net – Global Network of Intelligent Industrial Maintenance Web Services, In: IASTED International Conference on Databases and Applications (DBA 2004), Innsbruck, Austria, 17-19 February, 2004 (to appear).
25. Khriyenko O., Kononenko O., Terziyan V., OntoEnvironment: An Integration Infrastructure for Distributed Heterogeneous Resources, In: IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN 2004), Innsbruck, Austria, 17-19 February, 2004 (to appear).
26. 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 (to appear).
27. Basex Inc., "Executive report: Clerity for Enterprise Knowledge Sharing", January 2002.
28. Pirttioja T. "Agent-Augmented Process Automation System", Ms. Thesis, Helsinki University of Technology, Department of Electrical and Communications Engineering, Espoo, November, 7, 2002.
29. Yang B. and Garcia-Molina H. "Improving Search in Peer-to-Peer Networks", in Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS'02), IEEE 2002.
30. "Automation": Metso Automation's customer magazine, 3/ 2002: pp. 5-10.
31. Sarginson M., "Online instrumentation monitoring, impact on safety, uptime and process profitability", Metso Automation, Field Control Systems Committee, Exhibition 2002.
32. FIPA TC Communication. "FIPA ACL Message Structure Specification", SC00061G, Geneva, Switzerland, March, 2003.
33. Official website of the Protege 2000 Ontology Engineering Environment, http://protege.stanford.edu/, last accessed 12th Dec 2003.
34. Neles FieldBrowser, Predictive Maintenance made easy, http: //www.metsoautomation.com/, last accessed 12th Dec 2003.
35. W3C Recommendation, "SOAP Version 1.2, Part 0: Primer", http://www.w3.org/TR/soap12-part0/, 24 Jun 2003, last accessed 12th Dec 2003.
36. Siebes R. "Peer-to-Peer solutions in the Semantic Web context: an overview", EU-IST Project IST-2001-34103 SWAP, Vrije Universiteit Amsterdam, October 30th, 2002.