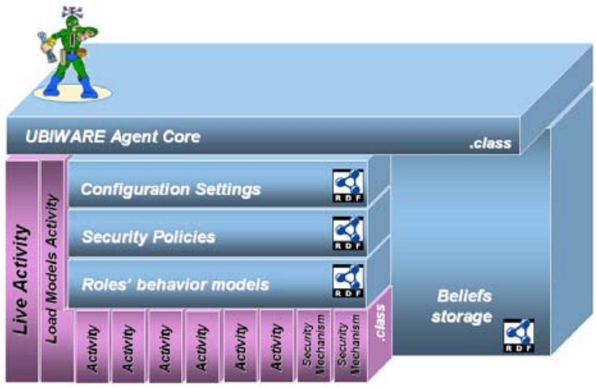
UBIWARE TEKES PROJECT – ANNUAL REPORT (2008-2009)

TEKES

UBIWARE Project

Year 2008-2009



Annual Report

Industrial Ontologies Group

Agora Center, University of Jyväskylä

metso automation shared services · FINGRID · ECURAS HANSA ECURAS OY · inno-W · NOKIA Connecting People · University of Jyväskylä

May, 2009

Jyväskylä, Finland

Table of Contents

Project's webpage: *http://www.cs.jyu.fi/ai/OntoGroup/UBIWARE_details.htm*

Group's website: *http://www.cs.jyu.fi/ai/OntoGroup/*

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

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

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

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

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

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

1 Introduction

Project Motivation

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

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

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

A major problem is inherent *heterogeneity* in ubiquitous computing systems, with respect to the nature of components, standards, data formats, protocols, etc, which creates significant obstacles for interoperability among the components of such systems. Semantic Web technologies are viewed today as a key technology to resolve the problems of interoperability and integration within heterogeneous world of ubiquitously interconnected objects and systems. The Internet of Things should become in fact the *Semantic Web of Things*[1]. Our vision for this project subscribes to this view. Moreover, we believe that Semantic Web technologies can facilitate not only the discovery of heterogeneous components and data integration, but also the behavioral control and coordination of those components.

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

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

Project Goals

This project intends to bring the Semantic Web, Distributed AI and Human-Centric Computing technologies to the Ubiquitous Computing domain, especially its industrial cluster. It aims at designing a new generation middleware platform (UBIWARE) which will allow creation of *self-managed* complex industrial systems consisting of mobile, distributed, heterogeneous, shared and reusable components of *different* nature. Those components can be smart machines and devices, sensors, actuators, RFIDs, communication systems and networks, web-services, software, information systems, humans, models, processes, organizations, etc. Such middleware will enable various components to automatically discover each other and to configure a system with complex functionality based on the atomic functionalities of the components.

We believe that tasks of automatic integration, orchestration and composition of such complex systems will be impossible with centralized control due to the scalability issue. Therefore, the components should be to a certain degree autonomous, proactive, and goal-driven. In other words, utilization of the agent technologies is needed to enable flexible communication and coordination of the components. Interoperability among the components requires use of metadata and ontologies. As the amount of components can grow dramatically, without their ontological classification and (semi- or fully-automated) semantic annotation processes, the automatic discovery will be impossible.

Project Stages

Research and development within this project will follow the following main directions (workpackages):

1. Core Distributed AI platform design (UbiCore);
2. Managing Distributed Resource Histories (UbiBlog);
3. Smart Ubiquitous Resource Privacy and Security (SURPAS);
4. Self-Management, Configurability and Integration (COIN);
5. Smart Interfaces: Context-aware GUI for Integrated Data (4i technology);
6. Middleware for Peer-to-Peer Discovery (MP2P);
7. Industrial cases and prototypes.

UBIWARE will require the reliable core platform to enable semantics-based proactivity and coordination of the components (workpackage WP1) and tools for managing and integrating distributed histories of the components (workpackage WP2). It will also require essentially new solutions towards security, service provisioning and information integration. Ubiquitous computing environment demands ubiquitous, yet flexible, security with respect to all kinds of interactions, based on well-defined and machine-readable policies (workpackage WP3). In UBIWARE, there is a need for new solutions towards self-management, configuration and integration of the components (workpackage WP4) and flexible semantic interfaces to deliver the integrated data to different human users having different backgrounds (workpackage WP5). UBIWARE should allow resource discovery not only based on centralized registries but also on the peer-to-peer basis (workpackage WP6). Finally, the middleware should be tried on real industrial cases to evaluate the scientific concepts behind it and facilitate its further utilization (workpackage WP7).

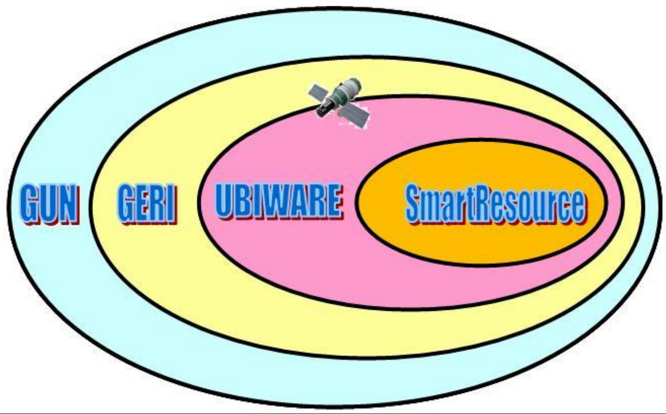
Workpackages 1 through 6 include both research and development tasks. The above tasks will be approached by combining various research methods with agile software development processes. This means that software prototypes will be iteratively developed during the whole project lifecycle based on real data, real needs and changing requirements of industrial partners. The result will be both the basic software tools for the UBIWARE platform and several industrial cases prototyped based on these tools. Prototypes of UBIWARE, integrating the work in the workpackages at different levels of their readiness, will be developed during each project year, as UBIWARE 1.0, UBIWARE 2.0 and UBIWARE 3.0, and reported through deliverables D1.3, D2.3, and D3.3, correspondingly. The status reports for the industrial cases will be collected in separate deliverables (one per year) D1.2, D2.2, and D3.2.

2 Project Background Concepts

This project is a next step of our research group towards the *Global Understanding Environment (GUN)* (Terziyan, 2003, 2005; Kaykova *et al.*, 2005a). The first step was done in the SmartResource project (2004-2006). Figure 1 depicts our research roadmap. A very general view on GUN is presented in Figure 2.

When applying Semantic Web in the domain of ubiquitous computing, it should be obvious that Semantic Web has to be able to describe resources not only as passive functional or non-functional entities, but also to describe their behavior (proactivity, communication, and coordination). In this sense, the word “global” in GUN has a double meaning. First, it implies that resources are able to communicate and cooperate globally, i.e. across the whole organization and beyond. Second, it implies a “global understanding”. This means that a resource A can understand all of (1) the properties and the state of a resource B, (2) the potential and actual behaviors of B, and (3) the business processes in which A and B, and maybe other resources, are jointly involved.

Global Understanding Environment (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.



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

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

UBIWARE – middleware for GERI

SmartResource – semantic technology, pilot tools and standards for UBIWARE

Figure 1 - The research roadmap towards GUN.

Main layers of GUN can be seen in Figure 2. Various resources can be linked to the Semantic Web-based environment via adapters (or interfaces), which include (if necessary) sensors with digital output, data structuring (e.g. XML) and semantic adapter components (XML to Semantic Web). Software agents are to be assigned to each resource and are assumed to be able to monitor data coming from the adapter about the state of the resource, make decisions on the behalf on the resource, and to discover, request and utilize external help if needed. Agent technologies within GUN allow mobility of service components between various platforms, decentralized service discovery, FIPA communication protocols utilization, and multi-agent integration/composition of services.

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

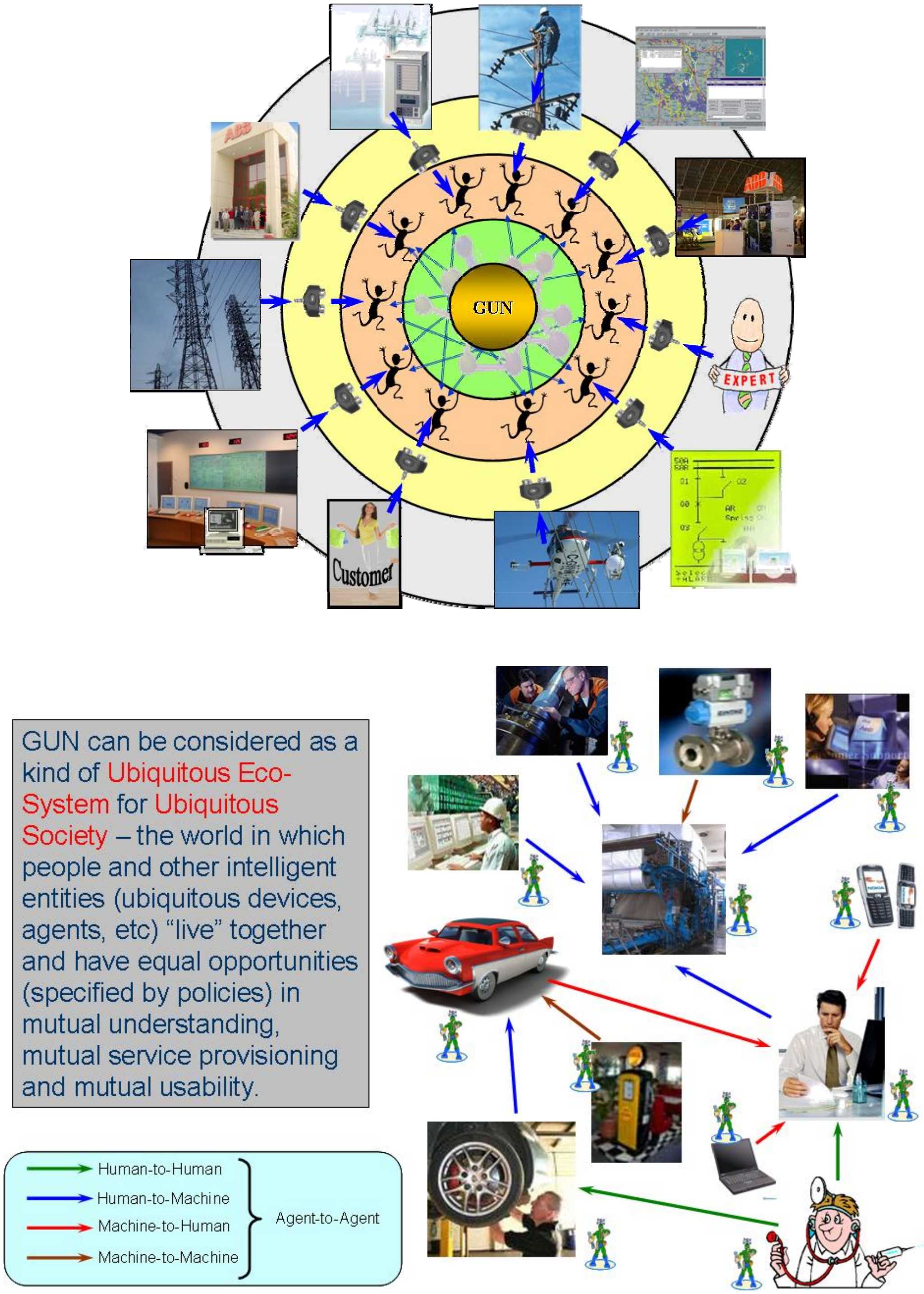


Figure 2 - The concept of Global Understanding Environment.

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

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

The second is the *General Proactivity Framework (GPF)* for automation and proactivity. GPF provides means for semantic description of individual behaviors by defining the Resource Goal/Behavior Description Framework (RgbDF). An implementation of GPF is supposed to include also an appropriate RgbDF-based domain ontology, an appropriate RgbDF engine and a family of “Semantic Adapters for Behavior” to provide an opportunity to transform data from a variety of possible behavior representation standards and formats to RgbDF and back. See more on RgbDF in (Kaykova *et al.*, 2005c).

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

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

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

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

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

According to a more global view to the Ubiquitous Computing technology:

- UBIWARE will classify and register various ubiquitous devices and link them with web resources, services, software and humans as business processes' components;
- UBIWARE will consider sensors, sensor networks, embedded systems, alarm detectors, actuators, communication infrastructure, etc. as "smart objects" and will provide similar care to them as to other resources.

Utilization of the Semantic Web technology should allow:

- Reusable configuration patterns for ubiquitous resource adapters;
- Reusable semantic history blogs for all ubiquitous components;
- Reusable semantic behavior patterns for agents and processes descriptions;
- Reusable coordination, design, integration, composition and configuration patterns;
- Reusable decision-making patterns;
- Reusable interface patterns;
- Reusable security and privacy policies.

Utilization of the Distributed AI technology should allow:

- Proactivity and autonomic behavior;
- Communication, coordination, negotiation, contracting;
- Self-configuration and self-management;
- Learning based on liveblog histories;
- Distributed data mining and knowledge discovery;
- Dynamic integration;
- Automated diagnostics and prediction;
- Model exchange and sharing.

Utilization of the Human-Centric approach enables us to consider humans in four possible roles (Figure 3):

- *Human as UBIWARE user* will get unique access to integrated and adapted services and information;
- *Human as UBIWARE service provider* will get support in online service provisioning and benefit as a servicing component in various business processes (e.g. a maintenance expert);
- *Human as UBIWARE resource* will be able to get online care from integrated distributed resources and services (e.g. monitoring the health of an employee);
- *Human as UBIWARE administrator* will be able to launch and configure UBIWARE for a particular task.

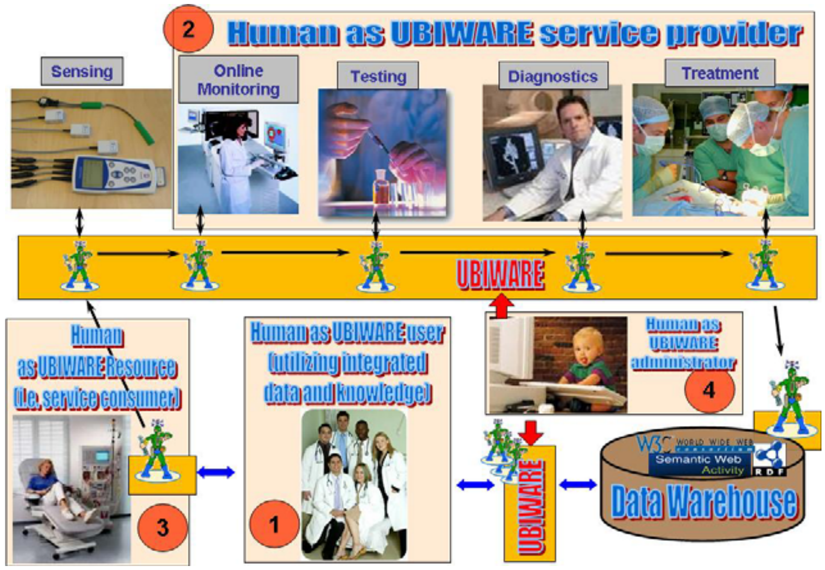


Figure 3 - Human in different roles in UBIWARE.

3 Project Results (Year 2008-2009)

The research results from the workpackages are reported through three integrating deliverables (one per project year):

1. *D2.1:* The central principles and tools of UBIWARE;
2. *D2.2:* Progress statuses of the industrial cases;
3. *D2.3:* UBIWARE Platform Prototype v.2.0.

3.1 The central principles and tools of UBIWARE – Deliverable 2.1

WP1: UbiCore

The main objective of the core platform is to ensure a predictable and systematic operation of the components and the system as a whole by

- enforcing that the smart resources, while might to have own "personal" goals, act as prescribed by the roles they play in a organization and by general organizational policies,
- maintaining the "global" ontological understanding among the resources, meaning 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

In the core platform that resulted from the research during Year 1, the behavior of an agent is prescribed by a set of behavioral models written in Semantic Agent Programming Language (S-APL). So far, it remained the full responsibility of the designer to ensure the absence of conflicts between different behavior models, e.g. roles, of one agent, as well as conflicts between different agents. During WP1's Year 2 (the *Deliberation* phase), the goal is to put more conflict-resolution intelligence into the platform, so that the agent would be able to resolve conflicts dynamically rather than rely on the absence of conflicts. The original plan for Year 2 included the following research questions:

- How to realize organizational policies as restrictions put on the behavior of individual agents?
- What mechanisms are needed for flexibly treating the potential (and likely) conflicts among the behavioral models (roles, policies) used by one agent?
- Is it possible and if yes then how to implements a mechanism so that agents could "see" what other agents are doing, in so creating the basis for coordination through observation in addition to traditional coordination through communication?

The work related to the first question has resulted in that the scope of the considered problem has been extended towards more general ontological coordination among autonomous agents. The traditional policies, such as of access control, are considered as a special case of coordination: an agent with an authority sets some restrictions on the behavior of other agents to avoid possible conflicts of interests. A more general case of coordination is where only when two autonomous behaviors conflict over the use of a resource some policy mechanism is applied. Such a resolution policy may dictate, for example, that the agent with a higher organizational authority gets the priority in using the resource.

The work related to the second question also resulted in somewhat extending the scope of the problem considered and shifting the focus – towards general-case agent's decision making when confronted with several options. Selecting among actions dictated by different roles is a special case of this.

WP2: UbiBlog

In UBIWARE, every resource is represented by a software agent. Among major responsibilities of such an agent is monitoring the condition of the resource and the resource's interactions with other components of the system and humans. The beliefs storage of the agent will, therefore, naturally include the history of the resource, in a sense "blogged" by the agent. Obviously, the value of such a resource history is not limited to that particular resource. A resource may benefit from the information collected with respect to other resources of the same (or similar) type, e.g. in a situation which it faces for the first time while other may have faced that situation before. Also, mining the data collected and integrated from many resources may result in discovery of some knowledge important at the level of the whole ubiquitous computing system. A scalable solution requires mechanisms for inter-agent information sharing and data mining on integrated information which would allow keeping the resource histories distributed without need to copy those histories to a central repository.

During WP2's Year 2 (the *Integration* phase), we work on the following question:

- How to realize the possibility of querying a set of distributed, autonomous, and, hence, inevitably semantically heterogeneous resource histories as they were one virtual database, i.e. how to collect and integrate needed pieces of information from distributed sources?

The problem of efficient data sharing, exchange and reuse within complex systems plays a key role for usability of the UBIWARE platform and its commercial success. In this workpackage we introduce a mechanism for distributed data management within the UBIWARE platform which allows platform users to build industrial business solutions.

In the research phase of WP2 we have introduced Ontonuts technology to tackle the problem of distributed querying in UBIWARE-based multi-agent systems. However, the theoretical foundations of the technology introduced go beyond the distributed querying problem and raise the question of semantic componentization in S-APL. The Ontonuts approach uses semantic annotation of the components' inputs and outputs in order to apply automated goal-driven planning. The backward chaining reasoning algorithm is used to build action plans of the agent. The technology also incorporates foundations of the General Adaptation Framework (GAF) developed in the SmartResource project. The GAF approach to transformation and mapping of the data and models was used in design of syntax of Ontonut annotations.

WP4: COIN

UBIWARE aims to be a platform that can be applied in different application areas. This implies that the elements of the platform have to be adjustable, could be tuned or configured allowing the platform to run different business scenarios in different business environments. Such flexibility calls for existence of a sophisticated configuration layer of the platform. All building blocks of the UBIWARE platform, i.e. software agents, agent behaviors, resource adapters, etc, become subject to configuration. On the other hand, a flexible system should have a long lifespan. Hence, the platform should allow extensions, component replacements, and component adjustments during the operation time. This work package aims at introducing configurability as a pervasive characteristic of UBIWARE and developing the technology which will systemize and formalize this feature of the platform.

During WP4's Year 2 (the *System* phase), we work on issues related to configuration of the system as a whole, through distributed decision making by agents representing the system components. WP will answer the following research questions:

- How a component of a system may realize the need for re-configuration of itself of the integral system, i.e. when the previous configuration of one or more components does not seem to work anymore?
- What mechanism are needed for (re-)configuration of the integral system through local decision making of and supported by communication between agents representing components of the system (i.e. with no central decision maker)?

In this part of the report we presented a solution to the self-configuration and self-manageability problem. We understand the term self-configuration as the ability of the multi-agent system to change the nature of its elements without any external intervention of the programmer. Instead the direct intervention, the system is able to configure itself based on a set of rules.

As a mathematical model we have chosen the DCOP model. The DCOP model contains reward/penalty functions that govern the process of the self-configuration. The goal is to find a configuration that has the biggest global utility (reward) based on the utility functions. We also explained the difference between algorithms seeking the global optimum and algorithms seeking a local optimum. From the group of algorithm seeking a local optimum, we described so-called k-optimal algorithms and explained what a k-optimum means. The report also contains a comparison of several complete and incomplete algorithms.

In the examples we showed step-by-step how a real-life problem can be converted into a DCOP problem and later solved using the UBIWARE platform. The first example showed how the DCOP model can be used to configure a group of WiFi access points in order to work efficiently. This example contained relatively simple binary constraints. The second example was dealing with the problem of best role selection. This problem was more complicated and it contained both unary and binary constraints.

WP5: 4I (FOR EYE) technology

This workpackage studies dynamic context-aware Agent-to-Human interaction in UBIWARE, and elaborates on a technology which we refer to as 4i (FOR EYE technology). From the UBIWARE point of view, a human interface is just a special case of a resource adapter. We believe, however, that it is unreasonable to embed all the data acquisition, filtering and visualization logic into such an adapter. Instead, external services and application should be effectively utilized. Therefore, the intelligence of a smart interface will be a result of collaboration of multiple agents: the human's agent, the agents representing resources of interest (those to be monitored or/and controlled), and the agents of various visualization services. This approach makes human interfaces different from other resource adapters and indicates a need for devoted research. 4i technology will enable creation of such smart human interfaces through flexible collaboration of an Intelligent GUI Shell, various visualization modules, which we refer to as MetaProvider-services, and the resources of interest.

WP5's Year 2 elaborates on probably the most important part of 4i vision, which can be called "context provision". Especially when considering a human, presenting information on a resource of interest alone is not sufficient - information on some "neighboring" objects should be included as well, which form the *context* of the resource. For example, a resource can be presented on a map thus shown in the context of objects which are spatial (geographic) neighbors of it. What is important is that in different decision-making situations, different contexts are relevant: depending on the situation the relevant neighborhood function may be e.g. physical spatial, data-flow connectivity, what-affects-what, similar-type, etc. The ability to determine what type of context in right one for the situation and collecting the information that forms the context of that type for a specific resource is central in 4i vision.

During WP5's Year 2 (the *Context-awareness* phase), therefore, the following research questions are to be answered:

- What should be the architecture of MetaProvider-services so that they will be able to effectively retrieve, integrate and deliver the context information both to for presenting to humans (in a visual form) and for agents' processing (semantic data)?

What should be the architecture of the Intelligent GUI Shell, so that it will allow situation-dependent selection of MetaProviders (i.e. different types of context) and cross-MetaProvider browsing and integration?

Sometimes user faces a need to find similar/close resources to the existing (selected) one. Thus, visualization of the resources in a context of their similarity/closeness becomes inherent functionality of the 4i(FOR EYE) Browser. We single out two types:

- *closeness* — closeness to resources of another different class;
- *similarity* — closeness to resources of a same class (can be considered as a particular case of *closeness* type);

Calculation of a distance measure is based on a set of preselected relevant/common properties and a measure of significance those properties on a distance measure. Each pair of resource classes has own set of relevant properties. But still, this attributes are configurable. User may reduce amount of properties (select just subset of them) to be considered in matchmaking algorithm and adjust/customize significance of the properties. If user does not provide detailed specification of the context, then matchmaking bases on all common contextual properties. As a result of a matchmaking algorithm, we have a matrix of distances between subject resource and other resources that helps us clearly show closeness via visual representation in GUI.

Approach of finding similar/close resources can be applied for closeness-based resource browsing. The idea is to visualize the resource in a context of its closeness to other resources based on just only one property selected by user. Then user can change the focus and select another resource to visualize it in a context of its closeness to other resources based on any its property. Such closeness can be searched among all relevant resources or among specific class of resources specified by user.

As the most used property value types we considering: *numerical* (numbers, dates) and *nominal* (textual strings (either defined on a finite or infinite set of possible values)) attributes. Different calculation procedures will be applied for different value types. Distance for numerical attributes can be calculated based on more or less well-known metrics; however, distance for nominal attributes (unstructured textual strings) is a challenge. For the automated distance measuring, row text fields should be additionally supported by list of strings (keywords/key-sentences). Distance measuring algorithm for the lists of strings can be invented and designed based on utilization of Google, WordNet, Wikipedia or other available online Web-services.

For the current implementation we concentrated on main five types of resource properties and invented correspondent distance measuring functions:

Text field types:

Type 1: Just a pure word/sentence. Additional contextual information for this field is its significance.

Type 2: Text field is presented by list of key words/sentences. Additional contextual information for this field is its significance.

Type 3: Text field is divided to the set of attributes and presented by correspondent list of values (words/sentences) of the attributes. In this case, the number of the attributes for certain text field should be defined and lists of possible (defined) values of the attributes should be defined and presented. In another words, it is defined amount of keywords, where each keyword is selected from a correspondent defined set of values. Additional contextual information for this field is the sets of values for each attribute (keyword) and the significance of the attributes, and as for all fields, significance of the field itself.

Number field: Just number that further will be normalized and compared. Additional contextual information for this field is its significance.

Interval field: Field presented by start and end point on a numerical axis. Distance measuring function for such interval field is based on a distance between the centers of the intervals and the lengths of them. Additional contextual information for this field is the significance of these two main parameters, and as for all fields, significance of the field itself.

3.2 Progress statuses of the industrial cases – Deliverable 2.2

WP7:

The objective of this workpackage is to trial UBIWARE on real industrial cases. This has two major goals for such case studies. The first goal is to evaluate the scientific concepts behind UBIWARE and to find problems and issues in UBIWARE that would otherwise be overlooked. The second goal is to facilitate the further utilization of UBIWARE in the industry. Several specific cases, proposed by the industrial partners, are analyzed, designed and prototyped based on the UBIWARE platform. The reasons for prototyping are the same: to identify issues in UBIWARE that would get overlooked if the work was only theoretical and thus abstract, and to demonstrate the benefits of UBIWARE in a tangible way so to facilitate future industrial adoption.

There are four industrial cases, those of Fingrid, Inno-W, Metso Automation and Nokia. During the Year 2, with respect to all four cases the task is developing a full prototype application: connecting to additional relevant resources and extending the interactions between them towards a sufficiently elaborated application.

Fingrid case:

With respect to the UBIWARE approach and platform, Fingrid's main area of interest is in organizing smart data management related to the events/alarms which company gets from their control systems. Existing systems do not provide many possibilities for managing this data beyond storing it to a time-series log, and browsing it with some filtering possibilities. A wish is to that the data should get flexibly accessible, integrated with other related data, and possibilities should be provided for producing generalizing reports to the power system operation and asset management persons.

Fingrid has the following two databases that were so far the objects of interest in the UBIWARE's Fingrid case:

- **Event History database**: **Eventlog** (Oracle) in the office environment, to which data is automatically replicated from SCADA's event history database. A record in this database contains such information as the time of the event, event class, access area, substation ID, device ID, the state of the object, and some other.
- **Elnet database** (Oracle) that stores information about all the equipment, including circuit-breakers, disconnectors, transformers, capacitors, and other. A record in this database contains such information as device group, device ID, ownership (Fingrid or external), and other.

The unique device ID present in both databases enables join queries.
The architecture of the Fingrid prototype is sketched in Figure 4. There are three agents in the prototype.

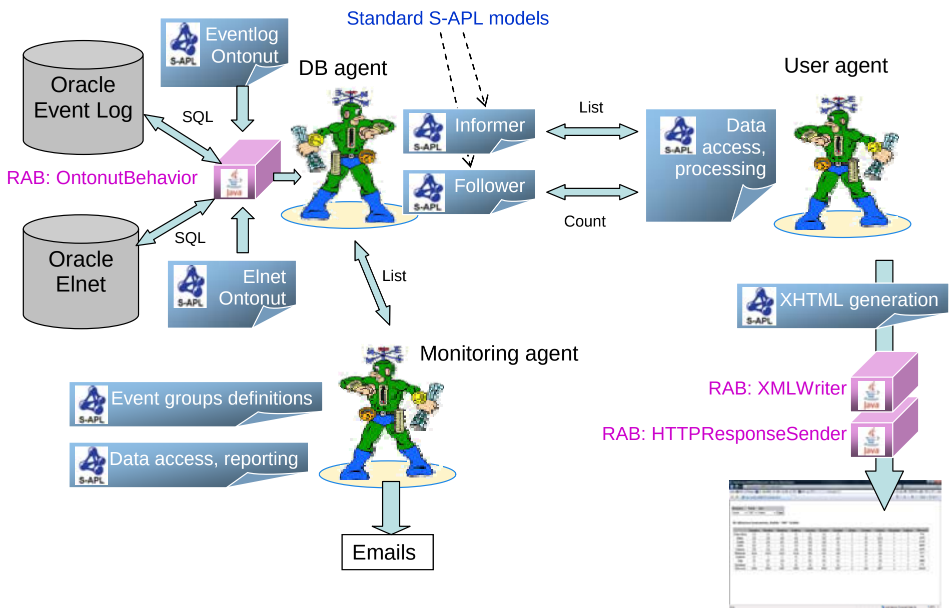


Figure 4 – Architecture of Fingrid prototype.

DB agent is responsible for interfacing with the databases. Implementation of this agent is based on UBIWARE's **ontonuts** approach. With this approach:

- DB agent receives from other agents queries that are formulated semantically and encoded using S-APL. The data is sent back to the requestors also in a semantic S-APL form.
- The databases (relational, non-semantic) are provided each with an *ontonut*, which is a description of the database schema that is sufficient for translating between S-APL semantic queries and SQL as well as between database responses and a needed semantic form.
- Reusable java component OntonutBehavior takes care of generating SQL and translating responses. If S-APL query concerns both databases, OntonutBehavior two generates SQL sub-queries and cross-joins the results.

It is notable that given that the interface of DB agent towards other agents is provided by standard S-APL models *Follower* and *Informer*, DB agent does not have any single line of code (either Java or S-APL) that would be written just for it and is non-reusable. The only tailored elements are ontonuts, which are declarations, not behavioral code.

User agent is responsible for providing XHTML interface to a human user. User agent receives user queries, interacts with DB agent, and presents the data to the user.

Monitoring agent is an autonomously operating agent which is responsible for continuously checking the new events appearing in the Eventlog database and sending email notifications.

From the operational point of view, the present prototype implement the three functions:
Function 1. *Equipment alarms:* This function allows a user to either count the number of R1 category events or to retrieve all such events from the Eventlog.

Function 2. *Operation counts:* The number of operations is counted for all circuit-breakers and disconnectors owned by Fingrid. One operation is one open - close cycle, so an operation is counted from by "Auki" (open) events.

Function 3. *Event groups:* This function adds pro-activity to the application in the sense that the application includes an agent that is autonomously:

- Checks for new events in the Event History database;
- Identifies if any of new events fall under the scope of some defined human "job responsibilities";
- Notify by e-mail the persons in charge about the event(s).

Inno-W case:

The goal of the Inno-W Company case is to build an Idea Browser that provides functionality to discover similar ideas/proposals/projects in user-defined context, to calculate and visualize similarity/closeness of ideas with GUI tool. Such a case requirement has the best match with workpackage WP-5 and realized based on 4I (FOR EYE) Browser.

Idea Browser is based on general 4I Browser architecture. The main common Interface part – 4I GUI Shell, performs communication with resource repository and repository of visualization contests. Shell provides resource search functionality, presents resources properties to the user, provides selection of resource visualization contexts and visualization modules (MetaProviders). At the same time, Shell provides all necessary data for MetaProviders. MetaProvider performs visualization function: depending on realization collects all necessary data and visualizes resource/resources in context dependent way. Considering the MetaProvider that presents resources in a context of their closeness to the selected one, certain distance measuring calculation is performed before visualization phase.

Comparison between the resources is performed based on common properties. Current implementation supports just five types of the parameters (properties):

Text field types:

Type 1: Just a pure word/sentence. Additional contextual information for this field is its significance.

Type 2: Text field is presented by list of key words/sentences. Additional contextual information for this field is its significance.

Type 3: Text field is divided to the set of attributes and presented by correspondent list of values (words/sentences) of the attributes. In this case, the number of the attributes for certain text field should be defined and lists of possible (defined) values of the attributes should be defined and presented. In another words, it is defined amount of keywords, where each keyword is selected from a correspondent defined set of values. Additional contextual information for this field is the sets of values for each attribute (keyword) and the significance of the attributes, and as for all fields, significance of the field itself.

Number field: Just number that further will be normalized and compared. Additional contextual information for this field is its significance.

Interval field: Field presented by start and end point on a numerical axis. Distance measuring function for such interval field is based on a distance between the centers of the intervals and the lengths of them. Additional contextual information for this field is the significance of these two main parameters, and as for all fields, significance of the field itself.

To visualize the resource closeness/similarity in the best way, we decided to put the resources on a spiral that lies on a surface of the cone (see Figure 6). Additionally, we provided a possibility to rotate the cone to find the best view point (see Figure 5).

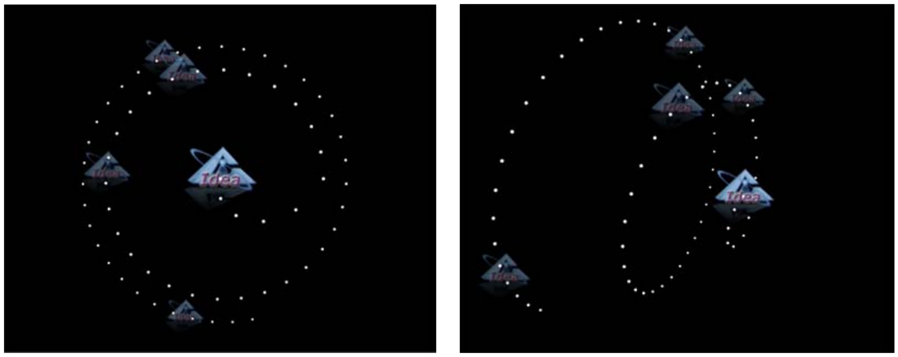


Figure 5 – spiral-based resource closeness visualization.

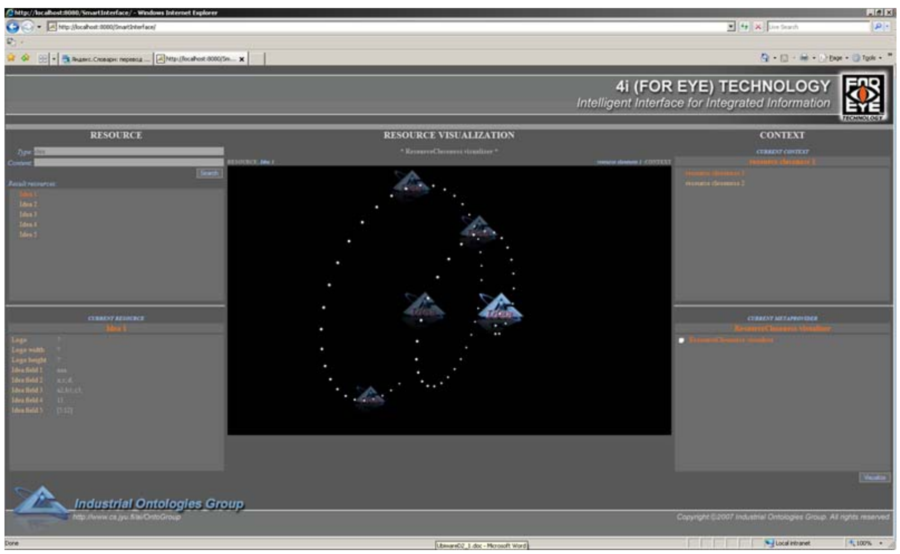


Figure 6 – resource closeness visualization in 4i (FOR EYE) Browser.

Metso case:

Metso industrial case was selected to be a test bed for a research and development within the WP2 (Distributed Querying and Integration) because of its “distributed nature” and big amounts of data in storages, that can not be collected in one place.

We integrate event flow data (events from monitoring and diagnostic systems) together with the structural and design data to provide a convenient assistant tool for an expert in diagnostics. In other words, ease the access to the relevant information needed for decision making.

We have specified a set of sources for integration:

- Alarm messages collected in RDF format (messages have been collected during two years)
- A sample of the DPM database (performance of each node of the paper machine)
- A sample of the Diary database (events handled and documented by factory workers)
- An excel sheet with the data about causticizing part of the plant from DNAExplorer.

The sources mentioned above are not fully integrated, but the most significant parts (derived from the use case) are semantically adapted. “Semantically adapted” means the description within the domain ontology and development of components, that represent the actual data sources as a virtual memory. I.e. the data is not fully transformed into S-APL, but it is annotated to answer the semantic queries instead.

The application consists of three main parts: Flash-based web application, Dispatcher servlet and Ontonuts agent. The whole process of communication is displayed in Figure 7.

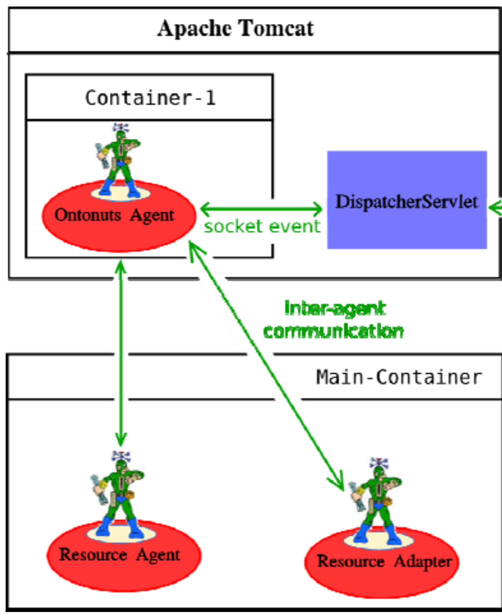


Figure 7 – The architecture of the prototype.

The flash-based application is running in the user's browser and communicates with the Dispatcher servlet via HTTP protocol. The whole communication is done asynchronously using the AJAX technology. The flash-based application is querying the servlet and the servlet dispatches the query to the Ontonuts Agent. When the query result is produced, it is sent back to the flash application. This is done every time the user initiates the query.
The communication between the Dispatcher servlet and the Ontonuts agent is done trough socket events. Ontonuts agent is running in a separate agent container (Container-1). The agent performs planning and executes sub-queries. For this purpose it accesses other resources (relational database, XLS sheets, sesame repository, etc.). For this reason it may further be connected to other agents, which act as resource adapters.

Metso Industrial case has been developed as an example usage scenario that demonstrates the applicability of the UBIWARE platform to corporate distributed search needs. The software presented here is easy to adapt to other problem domains, therefore future research directions will include automated configuration of the interface in accordance with the ontology provided. The agent interaction within the application is domain independent and the dependent parts are covered with the Ontonut definitions, which bind domain-dependent resources to the platform.

The Ontonuts engine is used within the application to create virtual data storage for an agent that answers user queries; however the applicability of this platform feature is not limited to querying. In the future platform releases we will improve the functionality of the engine to demonstrate its full potential.

Nokia case:

Mobile devices can now run web servers. However, the content inside the mobile devices is currently not accessible from outside (and cannot be indexed by Google). The content needs to be made public and flexibly searchable from outside depending on the access rights of the searcher.

Lots of research has been done how to index content and make it searchable on servers. But research how to search real-time from multiple servers organized into a social network is still very young. Nokia Research Center case advances social network search area and possibly generates a global scale search service.

1. UBIWARE project selected Last.fm[2] social network site as a source for social network simulation data to get information how music community is organized. Last.fm social network site was crawled with data of 177000 social network users including the top songs the users are listening and the associated albums.
2. Last.fm data is used as a data for Peer-to-Peer Realm simulator[3]. Parts of Peer-to-Peer Realm network simulator were reimplemented to include features for social network simulations and importing the whole Last.fm crawl to P2PRealm is under construction:
3. To study how different search algorithms work on a community of music listeners we need distributed algorithms for discovering local social network topology of each user. Topology awareness algorithm solving this problem has now been specified and implemented to P2PRealm. Next step is to implement social network search algorithm, which provides the first simulation results on Last.fm data.
4. Music listener should be able to get new and interesting music to his mobile device automatically and also be able to search music depending on his interests. This kind of functionality is planned to be provided via mobile social P2P prototype. User interface for such prototype is under design. With the prototype it will be possible to demonstrate real-time searching of relevant new content from other mobile phones (including also music content).
5. Social network search scenario can be further extended into a future information network. In addition to social aspect, this information network also supports searching based on location information and different interest areas of the users. Searching is a feature that should be actively done only by certain users and providing relevant information automatically to other inactive searchers would increase the speed of information flow significantly. Such functionality has been sketched as a part of the future information network.

The idea of the future information network is to provide means for users of the system to execute searches based on different distance measures and thus obtain information via traditional keyword search or even without any keywords depending on the density of URL pointers within the close neighborhood of the searcher. The user can learn of information that he/she does not know even exists (and therefore the user cannot search for it) by finding out the URLs that his/her social neighborhood has found interesting. Also the user can make his/her browsing information

[2] Last.fm – Discover new music with free internet radio and the largest music catalogue online, www.last.fm
[3] P2PRealm – Peer-to-Peer Network Simulator, kotilainen.eu/papers/P2PRealm.pdf

available to others and fully customize what parts of his/her personal device will be made accessible to other. This creates virtual representations of the users of the information network leading to a virtual reality experiences while using the information network. It is also expected that such kind of a network cannot be implemented in a centralized manner because of the large amount of information leading to bandwidth bottlenecks.

3.3 UBIWARE Platform Prototype v.2.0 – Deliverable 2.3

WP1: UbiCore

During the 2nd project year the core UBIWARE platform has become more mature. The platform performance has been significantly improved by introducing indices within the internal belief storage model. Also a number of bugs were fixed; some of them were critical, i.e. caused malfunctioning of scripts and behaviors. From the functional point of view the changes were minor, yet some of them bring additional important features.

Typically, an Ubiware-based application consists of several agents running within one or more JADE containers. The usual way of starting an application is to execute several scripts, one script per agent. This approach has two main disadvantages. Firstly, several scripts have to be run (each in separate window) which is not always possible if a number of agents grows. Secondly, if an agent crashes for some reason, it will not be restarted which will most likely make the application maintenance unreasonably complicated.

In order to overcome these limitations we decided to integrate Java Service Wrapper Community edition (http://wrapper.tanukisoftware.org/) to the Ubiware platform. It allows us to run a Ubiware-based application as a service. This has the following advantages:

- *The application is considered a service.* It is not needed to run several scripts. Also, if the application behaves as a Windows/Linux service, it may be started automatically every time the system is started.
- *The application is restarted every time it crashes.* There are several ways to define when and how the application should be restarted in case one of the agents dies.
- *The configuration file of the wrapper is platform independent.* This allows us to use one configuration file for several platforms. This basically means that if the application is run as a service on one platform (e.g. Windows) it can be run as a service also on another platform (e.g. GNU/Linux) without the need to change the configuration file.
- *Java Service Wrapper uses GNU GPLv2 license.*

The S-APL language as such has not undergone significant changes. The main construct that was added is a new operator for use in a new type of rules - an *Inference Rule.*

WP2: UbiBlog

The research we have conducted during the second year has resulted in the theoretical basis for the distributed querying of heterogeneous resources. We have introduced Ontonuts technology – an approach that allows us to represent physically distributed data in one place and the perform queries over this data as if it was integrated into one storage. The theoretical foundations of the technology can be found from the deliverable D2.1. In deliverable D2.3 we present an Ontonuts engine – a complex component that implements the above mentioned functionality.

The engine is an extension to the UBIWARE platform that allows platform user to easily connect external data sources and run distributed queries over them. The backward chaining algorithm was implemented to meet the platform-specific features and the language. The algorithm implementation is used in the planning of distributed queries. To support database connectivity, we have implemented a special type of Ontonut called Donut that provides additional functionality to the user when dealing with the relational data sources. Special attention was paid to the mapping and transformation (adaptation) of the external sources.

WP5: 4I (FOR EYE) technology

One of the 4I Browser enhancements concerns a smart and intelligent technique for automatic dynamic selection of a visualization context. The logic is based on a history of visualization contexts and resources that users have browsed/visualized previously. This context ranking technique allows us to sort a list of visualization contexts in more appropriate order for user and give him/her a hint for next logical step in though resource browsing process. Thus, it can become a smart search system that leads the user in proper direction/way.

There are a lot of contexts in which resources may be visualized. But, very often user faces a need to find similar/close resources to the initial one. Thus, we decided to include a visualization of the resources in a context of their similarity/closeness to the 4i(FOR EYE) Browser as its an inherent functionality.

One of the industrial cases that has been developed during the second project year and delivered in Deliverable 2.2 was focused on Idea similarity visualization and browsing. The main development in the direction of Deliverable 2.3 has been done during that period (see Deliverable 2.2):

- Distance Measuring Methods for five main resource description types and General Resource Distance Measuring Method;
- format and schema of similarity context related data;
- resource closeness/similarity Visualization component.

During the period after the previous checkpoint we add useful functionality to the system. Current implementation of the 4I GUI Shell supports visual configuration of resource similarity visualization context.

Visual configuration of resource similarity/closeness visualization context: Additionally to the previous development we added possibility to create new, delete and modify the similarity contexts. Such visualization context implies user specification of the resource properties significance and existence of additional contextual information for the resources properties (depending on their types).

To simplify the interface and make it more user friendly, we decided to consider the "*absolute significance*" of the resource fields as percentages from the full influence of the fields. In this case the sum of the fields' significances should be equal 100%. The same approach has been applied for the sub fields if there are any. For the "absolute significance" system supports two modes: *fully user controlled mode* and *mode with automatic recalculation of the significances.* Sometimes it becomes difficult to define the significance for all the fields in percentages, and user prefers to specify *“relative significance”* for the field/property. In this case user estimates the significance of each field/property by value from 0 to 100 separately. With the “relative significance” the absolute values do not make sense, only comparative differences if the values are taken into account. Further system itself transforms these values to the “absolute significance” and user can play with percentages later on if he/she wishes.

4 Achievements of the year

4.1 International cooperation

4.1.1 Scientific visits

During the second year of the project, one of the IOG members Artem Katasonov has been visiting USA with a purpose to disseminate results of the project and continue cooperation with research groups from MIT. The primary location was the University of Southern California, Los Angeles - Teamcore group lead by Prof. Milind Tambe. I also had a short visit to Massachusetts Institute of Technology (MIT) that included meetings with Prof. Tim Berners-Lee and Dr. Lalana Kagal from MIT Computer Science and Artificial Intelligence Laboratory (CSAIL), Prof. Edmund Schuster from MIT Data Center Program at the Laboratory for Manufacturing and Productivity. Edmund Schuster and colleagues at MIT Data Center Program work on interoperability in various industries through semantic data. We have been in contact with Data Center for long, but this was the first face-to-face meeting. Edmund has been familiarized with progress of Ubiware. He reported that now he understood better the value of Ubiware approach and even more ready to collaborate. Possibilities for exchange are discussed.

Another short visit was to the University of California, Berkeley. Artem had an extensive meeting with Prof. John Canny and a short one with Prof. Joseph Hellerstein, both from the Computer Science Division. Prof. Canny is an old colleague while Prof. Hellerstein is a new contact (work on declarative networking). Prof. Hellerstein and Prof. Canny have been familiarized with Ubiware work and found it interesting. Prof. Canny has described the current projects of their group. Interest shown for utilizing Ubiware experiences in the HealthMonitor project -organizing health-related measurements collected with an on-body wireless sensor device.

4.1.2 Project preparation

IOG has been prepared EU FP7 project **PRIME–II** - Proactive Inter-Middleware for Integrating Enterprise Systems into the Internet of Things. The original idea of the PRIME project as the main innovation supposed to introduce the concept of a "inter-middleware" (meta-middleware, 2nd order middleware). A "middleware" has been considered in a very general sense, as a kind of artificial world (ecosystem) into which the resources can register and get support for communication, collaboration, coordination, etc. among each other. There exists quite a lot of such middleware(s) already for various resources and purposes and will be more, of course, in future.

The innovative view of PRIME adds a recursion: each middleware is also a resource and the existing set of middleware(s)-resources may also need an artificial world (2nd order middleware) into which the resources-middleware(s) can register and get support for communication, collaboration, coordination, etc. among each other.

So what? The hope is: if the Inter-Middleware will be properly designed, this will allow to the resources (registered in one ecosystem) to communicate, collaborate, coordinate, etc with the resources registered in another ecosystem. Thus, by having just one entry point to some ecosystem (middleware) and corresponding interface, a resource will be able to utilize at least some services from other ecosystems.

Inter-Middleware will be really "beautiful" solution for the future Internet of things and it will have a lot of business opportunities.

We have been cooperating with a number of partners:

- IBM-Ireland, *Ireland*
- TELECOM SudParis, *France*
- National Univ. of Ireland, Galway, *Ireland*
- VTT, *Finland*
- University of Coimbra, *Portugal*
- Inno-W Oy, *Finland*
- Sapienza SL, *Spain*
- T.I.M.E. SRL, *Italy*
- Menta Networks, *Israel*

4.2 Awarded degrees

This year two IOG members have got following degrees:

- *Arnim Bleier* has obtained Master degree at the University of Osnabrück on September 30, 2008.
- *Oleksiy Khriyenko* has been awarded the degree of Doctor of Philosophy. Scientific postgraduate studies have been fulfilled in the field of Mathematical Information Technology. The dissertation "Adaptive Semantic Web based Environment for Web Resources" has been defended at its public examination on December 13, 2008. PhD Evgeny Osipov (Luleå University of Technology, Sweden) acted as the opponent and professor Timo Tiihonen (University of Jyväskylä) as the chairman.

5 Further Development

The year 3 of the project will, building on the created base, study more advanced topics, realize them in the UBIWARE platform, and in so progress towards the achievement of the general UBIWARE goal.

It was decided not to perform the work at the WP3 and WP6 during the third project year due to limitation in resources and research and develop the most important parts during the year. Therefore, project Year 3 integrates results from WP1&2, WP4, WP5 and WP7.

Work in this project is divided into five workpackages which are running in parallel:

1. Core agent-based platform design (WP1);
2. Managing Distributed Resource Histories (WP2);
3. Self-Management and Configurability (WP4);
4. Context-aware Smart Interfaces for Integrated Data (WP5);
5. Industrial cases and prototypes (WP7).

WP1: UbiCore

During WP1's Year 3 (the *Coordination* phase), we will investigate the approaches where behavioral S-APL models are used not only as prescriptive tool (i.e. loaded by agent to act based on them), but also as descriptive tool - accessed by other agents to e.g. understand what to expect from or how to interact with the agent in question. WP1 will attempt to answer the following research questions:

- How to enable agents to flexibly discover each other, based both on the roles played and on particular capabilities possessed.
- What would be concrete benefits of and what mechanisms are needed for accessing and using a role's script by agents who are not playing that role but wish to coordinate or interact with an agent that does?

The WP tasks for the Year 3 are the following:

Task T3.1_w1 (research): Answers to the questions above, and other, if appear in the process, related to the agents' semantic interaction based on each other's behavior models. Design of the core inter-agent discovery and coordination mechanisms.

Task T3.2_w1 (development): Incorporating the research findings to the UBIWARE prototype.

The tasks of this workpackage will be merged with the WP2, the argumentation for such a confluence is given below.

WP2: UbiBlog

In UBIWARE, every resource is represented by a software agent. Among major responsibilities of such an agent is monitoring the condition of the resource and the resource's interactions with other components of the system and humans. The beliefs storage of the agent will, therefore, naturally include the history of the resource, in a sense "blogged" by the agent. Obviously, the value of such a resource history is not limited to that particular resource. A resource may benefit from the information collected with respect to other resources of the same (or similar) type, e.g. in a situation which it faces for the first time while other may have faced that situation before. Also, mining the data collected and integrated from many resources may result in discovery of some knowledge important at the level of the whole ubiquitous computing system. A scalable solution requires mechanisms for inter-agent information sharing and data mining on integrated information which would allow keeping the resource histories distributed without need to copy those histories to a central repository.

During WP2's Year 1 (the *Sharing* phase), needed mechanisms were designed for effective and efficient sharing of information between different agents, e.g. representing different resources. S-APL was used as the communication content language, which has enabled:

- One agent to query another agent for some information, using the query constructs similar to that of SPARQL but with even wider range of possible filtering conditions
- One agent to inform another agent, i.e. to proactively push some information of any complexity.
- One agent to request another agent to perform some actions, either an atomic behavior or a complex plan involving a set of rules and atomic of complex behaviors.

During WP2's Year 2 (the *Integration* phase), we have realized the possibility of querying a set of distributed, autonomous and semantically heterogeneous resource histories as they were one virtual database. We have introduced a new concept of *Ontonuts* and a corresponding technology. The Ontonuts where tailored to solve the problem of distributed querying in the first place. However, we have generalized the concept to the level of semantic capabilities, which allows us to use goal-based dynamic planning and execution of agent's actions. Therefore, the third year phase of the WP2 (Mining phase) as it was specified before would rather be an application case of the Ontonuts technology than a research topic. With respect to this fact we have decided to concentrate our efforts towards the fusion of the WP1 and WP2 3rd year objectives. Shortly, the WP1 objective for the 3rd year is to elaborate a mechanism for advertising of agent capabilities and role scripts (complex capabilities) amongst agents, whereas WP2 research targets data mining capabilities only. Therefore, it is reasonable to integrate WP1 and WP2 under one umbrella of Ontonuts technology which is targeting even more ambitious goal – to enable automated (re)planning and execution of semantically annotated agent actions including distributed data querying, data mining as well as distributed agent-to-agent servicing.

WP4: COIN

Self-management and configurability in UBIWARE can be seen from two points of view:

- Initial self-configuration
- Runtime self-configuration

Initial self-configuration is understood as the ability of the system to interconnect and configure its components based on a certain goal or policy specified by the user. After specifying the goal of the system, the system itself should be able to automatically choose proper agents and delegate proper roles to them. The result should be a system that is performing the task specified by the goal. The user does not have to provide the system with any code, only the goal is needed. The system will find the best configuration based on the goal specified and a domain specific ontology.

Runtime self-configuration is the ability of the system to adapt to the environment. Thanks to this ability the system is able to perform its task even if the circumstances change. The process of runtime self-configuration should be context-aware, ontology-driven and policy-based.

Based on the text above, we specify the following research questions. First two are related to initial self-configuration and the third question is related to runtime self-configuration.

- How can we find suitable agents to perform the task based on the goal specified by the user?
- How can we transform a goal into a set of scripts to be executed by these agents?

- Once the system is configured, how can we maintain this state even if the circumstances of the system change?

The WP4 tasks for the Year 3 are the following:

Task T3.1_w4 (research): Answer the questions above and other related questions that may arise during the process. Find the limitations of the proposed approach.

Task T3.2_w4 (development): Incorporate the research findings to the UBIWARE prototype.

WP5: 4I (FOR EYE) technology

This work package studies dynamic context-aware Agent-to-Human interaction in UBIWARE, and elaborates on a technology which we refer to as 4i (FOR EYE technology). From the UBIWARE point of view, a human interface is just a special case of a resource adapter. We believe, however, that it is unreasonable to embed all the data acquisition, filtering and visualization logic into such an adapter. Instead, external services and application should be effectively utilized. Therefore, the intelligence of a smart interface will be a result of collaboration of multiple agents: the human's agent, the agents representing resources of interest (those to be monitored or/and controlled), and the agents of various visualization services. This approach makes human interfaces different from other resource adapters and indicates a need for devoted research. 4i technology is new paradigm of GUI development. It enables creation of such smart human interfaces through flexible collaboration of an Intelligent GUI Shell, various visualization modules, which we refer to as MetaProvider-services, and the resources of interest.

According to the last discussion during the steering group meeting, we agreed to concentrate 3rd year project research on business issues, commercialization steps of the results. Now when we have clear vision of the idea, have the elaborated prototype of an initial idea, we are ready for the next valuable step during WP5's Year 3. This step will consist of two parts: elaboration of the general architecture of the product (necessary components, tools and utilization models) and commercialization part (business and market analysis, business models, promotion, distribution and etc.).

During WP5's Year 3 (the *Commercialization* phase), therefore, the following research questions are to be answered:

- What should be the general architecture of the product – 4I(FOR EYE) tool package so that it will be possible to build and further extend a different services based on the product? What are the requirements for the product, for the product components, what are the necessary modifications and the use cases of the product utilization?
- What are the commercialization and marketing steps?

The WP tasks for the Year 2 are the following:

Task T3.1_w5 (research): Answer to the questions above. Design of a 4I(FOR EYE) tool package and utilization business models.

Task T3.2_w5 (development): Incorporating the research findings to the current prototype, and bring it close to the real product.

There are three industrial cases, those of Fingrid, Metso Automation and Inno-W.

During the Year 3, with respect to all three cases the task is the same as for the Year 2:

Task T3.1_w7: Developing a full prototype application: connecting to additional relevant resources and extending the interactions between them towards a sufficiently elaborated application.

Detailed plans of the industrial cases will be presented after discussions and negotiations with the companies.

6 UBIWARE Project Publications (up to the April of year 2009)*

[1] Katasonov A., Terziyan V., Using Semantic Technology to Enable Behavioural Coordination of Heterogeneous Systems, In: V. Kordic (ed.), Semantic Web, IN-TECH Publishing, 2009, ISBN: 978-953-7619-33-6, 22 pp. (Book Chapter, to appear).

[2] Nagy M., Katasonov A., Khriyenko O. Nikitin S., Szydlowski M., Terziyan V., Challenges of Middleware for the Internet of Things, In: A. Lazinica (ed.), Robotics, Automation and Control, IN-TECH Publishing, 2009, ISBN: 978-953-7619-39-8, 24 pp. (Book Chapter, to appear).

[3] Kesäniemi J., Katasonov A., Terziyan V., An Observation Framework for Multi-Agent Systems, In: Proceedings of the Fifth International Conference on Autonomic and Autonomous Systems (ICAS 2009), April 21-25, 2009, Valencia, Spain, IEEE CS Press, 6 pp.

[4] Katasonov A., Terziyan V., Semantic Approach to Dynamic Coordination in Autonomous Systems, In: Proceedings of the Fifth International Conference on Autonomic and Autonomous Systems (ICAS 2009), April 21-25, 2009, Valencia, Spain, IEEE CS Press, 9 pp.

[5] Terziyan V., Zhovtobryukh D., Katasonov A., Proactive Future Internet: Smart Semantic Middleware for Overlay Architecture, In: Proceedings of the Fifth International Conference on Networking and Services (ICNS-2009), April 21-25, 2009, Valencia, Spain, IEEE CS Press, 6 pp.

[6] Nikitin S., Katasonov A., Terziyan V., Ontonuts: Reusable Semantic Components for Multi-Agent Systems, In: Proceedings of the Fifth International Conference on Autonomic and Autonomous Systems (ICAS 2009), April 21-25, 2009, Valencia, Spain, IEEE CS Press, 8 pp.

[7] Khriyenko O., Adaptive Semantic Web based Environment for Web Resources, In: Jyvaskyla Studies in Computing, PhD Thesis, Volume 97, Jyvaskyla University Printing House, 192 pp., December 13, 2008.

[8] Bleier A., A Framework for Market-Based Coordination in Multi-Agent Systems, MSc Thesis, University of Osnabrück, September 30, 2008.

[9] Terziyan V., Semantic Web Services for Smart Devices Based on Mobile Agents, In: D. Taniar (Ed.), Mobile Computing: Concepts, Methodologies, Tools, and Applications (6 volumes), IGI Global, November 2008, ISBN: 978-1-60566-054-7, Vol. II, Chapter 2.22, pp. 630-641.

* Papers are downloadable from http://www.cs.jyu.fi/ai/OntoGroup/UBIWARE_details.htm

[10] Terziyan V. and Katasonov A. (2008) Global Understanding Environment: Applying Semantic and Agent Technologies to Industrial Automation, In: Lytras, M. and Ordonez De Pablos, P. (eds) Emerging Topics and Technologies in Information Systems, IGI Global , 2009, ISBN: 978-1-60566-222-0, pp. 55-87 (Chapter III).

[11] Katasonov A. and Terziyan V. (2008) Semantic Agent Programming Language (S-APL): A Middleware Platform for the Semantic Web, In: Proc. 2nd IEEE Conference on Semantic Computing, August 4-7, 2008, Santa Clara, CA, USA, pp.504-511.

[12] Khriyenko O., Context-sensitive Visual Resource Browser, In: Proceedings of the IADIS International Conference on Computer Graphics and Visualization (CGV-2008), Amsterdam, The Netherlands, 24-26 July 2008.

[13] Katasonov A., Kaykova O., Khriyenko O., Nikitin S., Terziyan V., Smart Semantic Middleware for the Internet of Things, In: Proceedings of the 5-th International Conference on Informatics in Control, Automation and Robotics, 11-15 May, 2008, Funchal, Madeira, Portugal, ISBN: 978-989-8111-30-2, Volume ICSO, pp. 169-178.

[14] Terziyan V., SmartResource - Proactive Self-Maintained Resources in Semantic Web: Lessons learned, In: International Journal of Smart Home, Special Issue on Future Generation Smart Space, Vol.2, No. 2, April 2008, SERSC Publisher, ISSN: 1975-4094, pp. 33-57.

[15] Katasonov A. and Terziyan V. (2007) SmartResource Platform and Semantic Agent Programming Language (S-APL), In: Proceedings of the 5th Conference on Multi-Agent Technologies (MATES’07), September 24-26, 2007, Leipzig, Germany, LNAI 4687, pp.25-36.

[16] Terziyan V., Predictive and Contextual Feature Separation for Bayesian Metanetworks, In: B. Apolloni et al. (Eds.), Proceedings of KES-2007 / WIRN-2007, Vietri sul Mare, Italy, September 12-14, Vol. III, Springer, LNAI 4694, 2007, pp. 634–644.

[17] Khriyenko O., Context-sensitive Multidimensional Resource Visualization, In: Proceedings of the 7th IASTED International Conference on Visualization, Imaging, and Image Processing (VIIP 2007), Palma de Mallorca, Spain, 29-31 August 2007.

[18] Khriyenko O., 4I (FOR EYE) Multimedia: Intelligent semantically enhanced and context-aware multimedia browsing, In: Proceedings of the International Conference on Signal Processing and Multimedia Applications (SIGMAP-2007), Barcelona, Spain, 28-31 July 2007.

[19] Khriyenko O., 4I (FOR EYE) Technology: Intelligent Interface for Integrated Information, In: Proceedings of the 9th International Conference on Enterprise Information Systems (ICEIS-2007), Funchal, Madeira – Portugal, 12-16 June 2007.

[20] Salmenjoki K., Tsaruk Y., Terziyan V., Viitala M., Agent-Based Approach for Electricity Distribution Systems, In: Proceedings of the 9-th International Conference on Enterprise Information Systems, 12-16, June 2007, Funchal, Madeira, Portugal, ISBN: 978-972-8865-89-4, pp. 382-389.

[21] 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, ISBN: 978-972-8865-87-0, pp. 191-194.

[22] Naumenko, A., Srirama, S., Secure Communication and Access Control for Mobile Web Service Provisioning, In: Proceedings of International Conference on Security of Information and Networks (SIN2007), 8-10th May, 2007.

[23] Naumenko A., Semantics-Based Access Control in Business Networks, In: Jyvaskyla Studies in Computing, PhD Thesis, Volume 78, Jyvaskyla University Printing House, 215 pp., June 28, 2007.

[24] Naumenko A., Katasonov A., Terziyan V., A Security Framework for Smart Ubiquitous Industrial Resources, In: R. Gonzalves, J.P. Muller, K. Mertins and M. Zelm (Eds.), In: Enterprise Interoperability II: New challenges and Approaches, Proceedings of the 3rd International Conference on Interoperability for Enterprise Software and Applications (IESA-07), March 28-30, 2007, Madeira Island, Portugal, Springer, pp. 183-194.

[25] Naumenko A., SEMANTICS-BASED ACCESS CONTROL - Ontologies and Feasibility Study of Policy Enforcement Function, In: Proceedings of the 3rd International Conference on Web Information Systems and Technologies (WEBIST-07), Barcelona, Spain - March 3-6, 2007, Volume Internet Technologies, INSTICC Press, pp. 150-155.

Submitted or going to be submitted:

[26] Khriyenko O., Terziyan V., Similarity/Closeness-Based Resource Browser, In: Proceedings of the Ninth IASTED International Conference on Visualization, Imaging and Image Processing (VIIP-2009), July 13-15, 2009, Cambridge, UK, 7 pp. (submitted).

7 References

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

Kaykova O., Khriyenko O., Naumenko A., Terziyan V., and Zharko A. (2005b), RSCDF: A Dynamic and Context Sensitive Metadata Description Framework for Industrial Resources, In: Eastern-European Journal of Enterprise Technologies, Vol.3, No.2, pp. 55-78.

Kaykova O., Khriyenko O., Terziyan V., and Zharko A. (2005c) RGBDF: Resource Goal and Behaviour Description Framework, Proc. 1st International Conference on Industrial Applications of Semantic Web, Springer, IFIP, vol.188, pp. 83-99.

Khriyenko O., and Terziyan V. (2006), A Framework for Context-Sensitive Metadata Description, In: International Journal of Metadata, Semantics and Ontologies.

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

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

Terziyan V. (2005) Semantic Web Services for Smart Devices Based on Mobile Agents, In: International Journal of Intelligent Information Technologies, Vol. 1, No. 2, Idea Group, pp. 43-55.