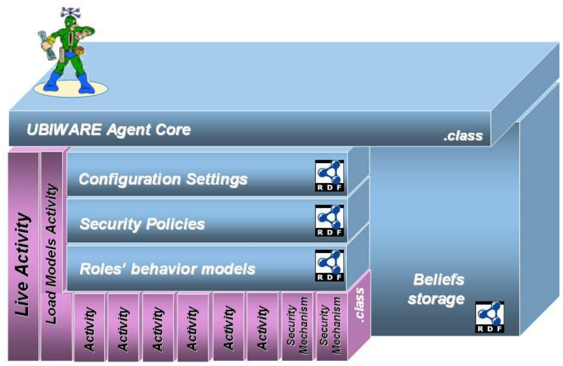
UBIWARE TEKES PROJECT – ANNUAL REPORT (2007-2008)

TEKES

UBIWARE Project

Year 2007-2008



Annual Report

Industrial Ontologies Group

Agora Center, University of Jyväskylä

ABB metso automation shared services FINGRID ECURAS HANSA ECURAS OY Inno-W University of Jyväskylä

May, 2008
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, see Section **Error! Reference source not found.** for details) 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).

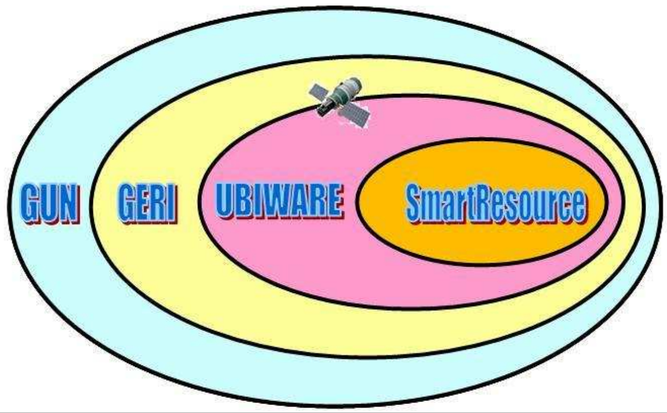
Workpackages 1 through 6 include both research and development tasks. The above tasks will be approached by combining various research methods with agile software development processes. This means that software prototypes will be iteratively developed during the whole project lifecycle based on real 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.

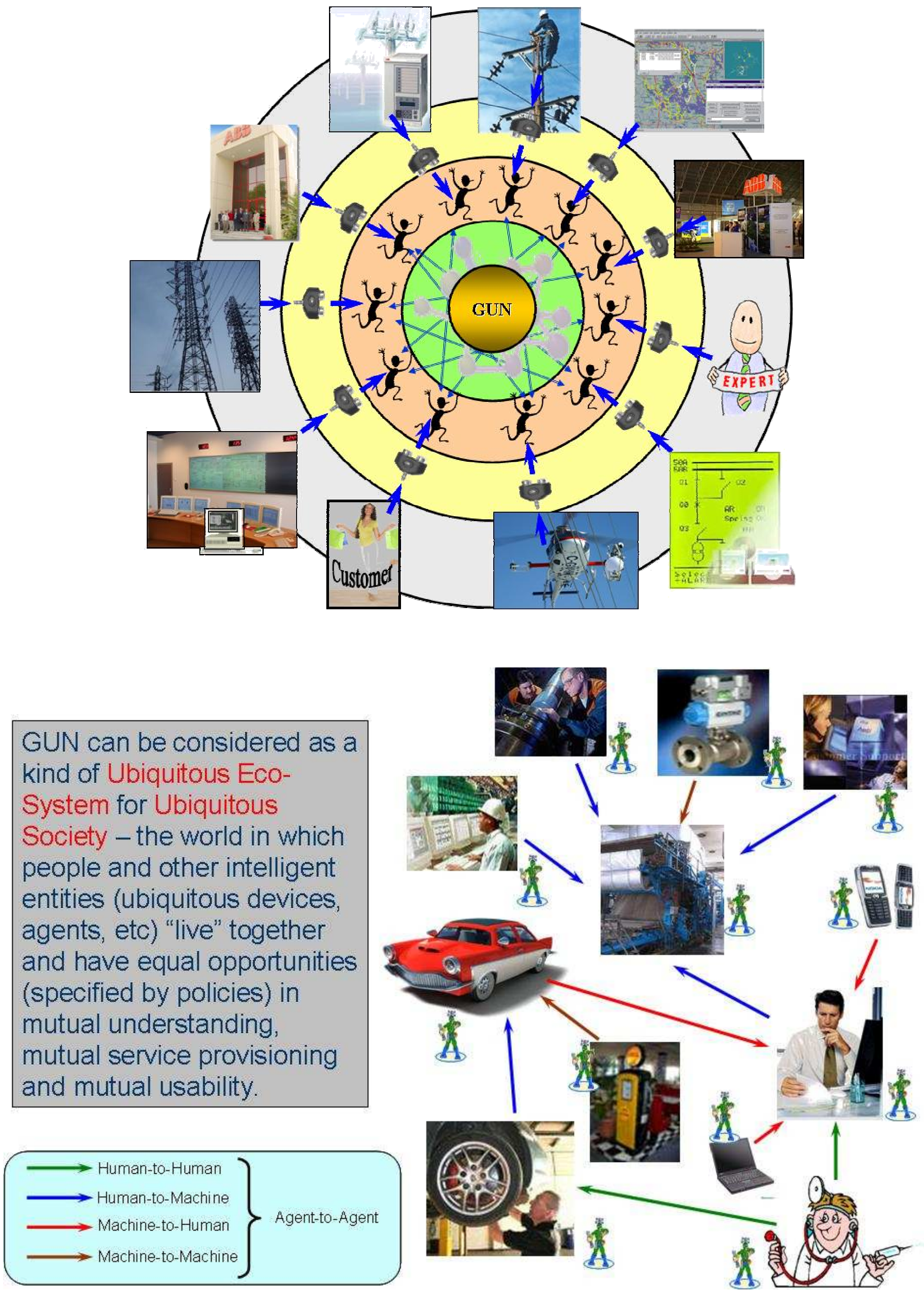


Figure 2 - The concept of Global Understanding Environment.

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.

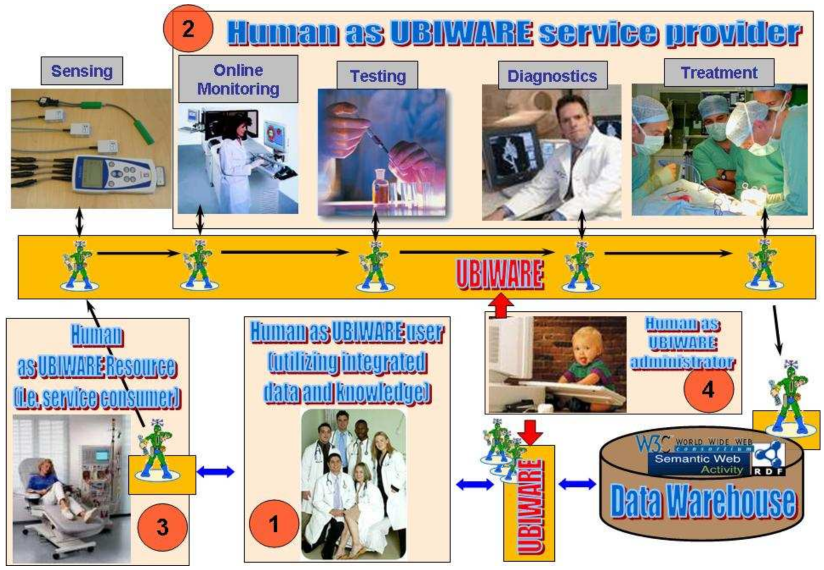


Figure 3 - Human in different roles in UBIWARE.

3 Project Results (Year 2007-2008)

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

1. *D1.1:* The central principles and tools of UBIWARE;
2. *D2.1:* Progress statuses of the industrial cases;
3. *D3.1:* UBIWARE Platform Prototype v.1.0.

The central principles and tools of UBIWARE – Deliverable 1.1

WP1: UbiCore

The main objectives of the UbiCore are the following. It has to give every resource a possibility to be smart (by connecting a software agent to it), in a sense that it would be able to proactively sense, monitor and control own state, communicate with other components, compose and utilize own and external experiences and functionality for self-diagnostics and self-maintenance. It has to enable the resources to automatically discover each other and to configure a system with complex functionality based on the atomic functionalities of the resources. It has to ensure a predictable and systematic operation of the components and the system as a whole by enforcing that the smart resources act as prescribed by their organizational roles and by maintaining the "global" ontological understanding among the resources. The latter 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

The part of the work in this work-package, reported in the deliverable, answered the following research questions:

- How the language for roles' scripts, developed in SmartResource project, has to evolve to enable the full spectrum of possibilities that is found in Agent Programming Languages (APLs) – to become, in addition to other benefits, a Semantic APL?
- How to implement the separation between a role's capabilities (individual functionality), and the business processes in which this role can be involved (complex functionality)?

As the main advantages of S-APL we highlighted the following:

- Simple model (triples in hierarchical contexts) that allows implementing any feature found in existing APLs.
- Expressive power is even greater than in existing APLs, because of full symmetry (everything is a belief): e.g. rules upon execution can add other rules of any complexity.
- Behavior specification is done using semantic predicates (e.g. implies, existsWhile)
    - Formally defined in an ontology.
    - Language is extensible with other such predicates.
- Reusable Atomic Behaviors and their parameters are also resources that can (and should) be ontologically modelled.
- So, there is *a basis* for sharing all 5 ontologies: External world, Mental states of agents, Properties of agents' bodies (available sensors and actuators), Input properties and Output properties (Bosse, T., Treur, J., 2000). Therefore, there is a basis for better understanding among agents with a goal of better coordination and collaboration among them.

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.

The part of the work in this work-package, reported in deliverable, answered the following research question:

- How to semantically markup the history of a resource in a system, in order to make it reusable for other resources and at the system-level?
- What mechanisms are needed for effective and efficient sharing of information between the agents representing different resources?

The general approach, which was selected in this work-task, is to avoid designing some special protocols and languages for inter-agent information sharing, but rather to reuse as much as possible the tools developed as part of the UbiCore (WP1).

Obviously, there is certain similarity in the following. On one hand, an agent always needs to query its own beliefs base in order to evaluate the left sides of its behavior rules in order to identify rules that are to be executed. On the other hand, when an agent asks another agent for some information, it, in a sense, queries the belief base of that other agent.

Our approach is therefore to design the external querying process so it would be almost the same as if the agent itself would query its belief base to check the conditions for executing a rule. This also means that we plan to use the Semantic Agent Programming Language (S-APL) not only as the means for prescribing the agents' behaviors, but also as the inter-agents communication content language (to be used instead of FIPA-SL or other languages of this type). The advantages of this should be obvious as the symmetry and expressive power in the UBIWARE platform will be maximized. The agents will be able to query each other not only for some facts (present or historical) about the external world (the domain) but also, for example:

- Query if the other agent knows a plan for achieving a certain goal,
- Query if the other agent knows a rule that should be applied in a particular situation.

We did not repeat the details about S-APL that have already been described in the WP1, and presented only additional information.

WP3: SURPAS

Globalization of the economy, global and intercultural value chains, large-scale industrial environments, cooperative systems for the international production, logistic and marketing could hardly be imagined without the rapid evolution of information and communication technologies (ICTs). Moreover, continuous advances of ICTs and their adoption in the industrial world have been guaranteeing improvement and efficiency of industrial technologies in the last decades. Recent advances in networking, sensor and RFID technologies, etc 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. However, the adoption of new ICTs in the traditional production industries, e.g. the process industry, the machinery industry, etc, is relatively slow. It is mainly because of the growing complexity of emerging ICTs, inadequate security infrastructures, and the fact that the research in ICTs usually focuses on the industries with a short cycle of innovations deployment, such as health care or banking, largely overlooking the needs of the production industries.

In response to these problems, this part of the deliverable focuses on the security challenges in UBIWARE. It presents the security threats, requirements, implications and access control measures needed for UBIWARE in the context of its industrial adoption. Industrial cases align our research results on UBIWARE, as such, with the real world needs and serves as a trigger and source of requirements for the research on security, particularly. We describe our long-term vision for the security and privacy management in emerging new types of environments, which we refer to as Smart Ubiquitous Resource Privacy and Security (SURPAS). SURPAS is mainly based on the advances in the Semantic Web, Multi-Agent Systems, and Ubiquitous Computing domains. Particularly, this part presents the SURPAS research framework which guides our research towards SURPAS. It is a consolidated formal system of research ideas and prototypes for the interoperable pro-active context-aware self-protecting security management. The main components of the SURPAS research framework are the conceptual semantics of security policies, functionality of security mechanisms, including functional semantics, algorithms, abstract architecture, and reference implementation, and adopting applications in different business domains (e.g. industrial maintenance, subcontracting management, smart house, etc).

The part of the work in this work-package, reported in deliverable, answered the following research question:

- Identifying, modeling, and assessment of security threats and requirements for the UBIWARE.
- Conceptual and functional semantics, algorithms and abstract architecture of SURPAS: Access Control.

We addressed the security threats in UBIWARE, presented analysis of security concerns, regarding industrial adoption of UBIWARE, and the SURPAS research framework. Also we gave a detailed description of the SURPAS conceptual and functional semantics, exemplified our research and development ideas using the industrial cases, presented conclusions and future research directions.

Conventional approaches to manage and control security seem to have reached their limits in new complex environments. These environments are open, dynamic, heterogeneous, distributed, self-managing, collaborative, international, nomadic, ambient, and ubiquitous. New generation middleware such as UBIWARE will significantly advance the industrial automation towards automatic discovery, composition, orchestration, integration, invocation, execution monitoring, and coordination of industrial resources. These advanced automation techniques target physical world objects and thus put security as the core need-to-be-addressed issue. We described our long-term vision for the security and privacy management in such complex environments, SURPAS. It aims at policy-based optimal collecting, composing, configuring and provisioning of security measures in multi-agent systems like UBIWARE. This part concentrates on the access control issues in SURPAS. Particularly, we analyzed the security implications of UBIWARE, presented the SURPAS research framework which guides our research towards SURPAS, the SURPAS conceptual semantics and the SURPAS abstract architecture.

There are an enormous number of targets for further work. They include ontology engineering for fundamental elements of security, elaborating architectures, designing new specific algorithms for the intelligent security policy management, developing reference implementations and, finally, adopting research ideas into practice in real-world industrial settings.

WP4: COIN

Autonomic components must be given certain degree of flexibility in order to make them more reusable and increase adaptability to new or changing environment. One of the key issues is to keep connectivity to the resources from the outer world (with respect to UBIWARE) connected via adapters. To keep resources always connected, we need to elaborate a framework for configurable adaptation, which will define the principles of the adapter configuration and change handling rules.

The notion of configurability and configuration is nowadays discussed in different application areas ranging from the Reconfigurable Computing[2] to Software Configuration Management[3]. Although, these topics seem to be relevant, they are quite different from the problem domain of the UBIWARE. In this paper we will narrow our scope to the configurability of software components, meaning that components have interfaces allowing changes to the

[2] Reconfigurable Computing - http://en.wikipedia.org/wiki/Reconfigurable_computing
[3] SCM - http://en.wikipedia.org/wiki/Software_configuration_management

components' behaviour. Furthermore, in order to illustrate the configurability in action, we select a specific component of the UBIWARE platform – an adapter, as an object of configuration.

The need for configurable software is dictated by the market demanding more and more adjustable and flexible solutions. There is a number of open source solutions available, such as Obix framework (OBIX). The framework offers an API to incorporate software configuration mechanisms into your application. It works with the XML-based portable configuration data, allows definitions of relationships amongst components, detects automatically changes to the configuration files and loads new settings. These solutions aim at the configurable initialization and deployment of applications and offer rather sophisticated APIs. The UBIWARE platform provides initial configurations as well, so these frameworks can be useful, however, due to autonomous nature of UBIWARE architecture, we need an instrument, that allows on-the-fly component (re-) configuration. For more sophisticated self-aware entities such as software agents, we need a mechanism for self-configuration.

The generic problem of configurability is a hot topic in a number of today's ICT areas. Reconfigurable hardware elements have brought new tasks to architects both on hardware and software levels. There are a number of on-going research activities in a field of reconfigurable and/or self-aware systems, for example SELFMAN (SELFMAN), CASCADAS (CASCADAS) and Deliver (Deliver).

The part of the work in this work-package, reported in deliverable, answered the following research question:

- Evaluating current approaches towards software configuration, configuration schemas and patterns. Developing a methodology for creating configurable resource adapters.

The need for configurable adaptation has appeared as a result of industrial cases analysis, which disclosed the problems of the static code in process industry. Minor changes in the business logic of information systems lead to maintenance breaks and involve a lot of human resources, particularly programmers for code maintenance, testers, system administrators for giving the access to the running platforms, and deployment, who launch the updated version as a product. The configurability changes the process with the anticipation of possible changes and making them a part of the functionality in the very beginning, thus giving more flexibility to the business process management. Most of the changes now go from the programmer's level to the level of business process configuration – a descriptive script-based business logic, which can be updated dynamically in a runtime. This allows the business process manager to handle a vast number of situations without annoying the programmers and provide fast changes in response to customers needs. The configurability can be used not only for a product customization, but also for a dynamic, on the fly support and maintenance.

WP5: 4I (FOR EYE) technology

Accordingly to Lyndon J.B. Nixon work (Nixon, 2006), as the current trends develop we expect to experience a future Web which will be media rich, highly interactive and user oriented. The value of this Web will lie not only in the massive amount of information that will be stored within it, but the ability of Web technologies to organize, interpret and bring this information to the user. And as usually, a graphical user interface is one of the important parts in performing of these processes. Media presentation is a key challenge for the emerging media-rich Web platforms. Several information visualization techniques have been developed in the last years due to the need of representing and analyzing the huge amount of data generated by several applications or made available through the World Wide Web. Previously, we had a deal with data visualization, precisely with a data format representation. Depending on a data format, whether it is a text, an image or a video, graphical user interface presents the data in certain way. On the next stage, small step has been done in visualization of object part-of relations, namely as a tree visualization. The second step was a step when we came to semantic definition of the objects, and have found a need to represent ontology concept tree and semantic graph (Yuxin et al., 2005). But it was just a step to semantics and ontology representation.

In following new technological trends, it is time to initiate a new stage of multidimensional resource visualization (visualization of resource properties, contexts of inter-resource communication and interaction) and a stage of semantic metadata-based visual browsing across resources.

The part of the work in this work-package, reported in deliverable, answered the following research question:

- General vision of 4i technology and its application in UBIWARE. Requirements for Human Adaptation.

Now, when human becomes very dynamic and proactive resource of a large integration environment with a huge amount of different heterogeneous data, it is quite necessary to provide a technology and tools for easy and handy human information access and manipulation. Semantically enhanced context-dependent multidimensional resource visualization provides an opportunity to create intelligent visual interface that presents relevant information in more suitable and personalized for user form. Context-awareness and intelligence of such interface brings a new feature that gives a possibility for user to get not just raw data, but required information based on a specified context. Now, when unlimited interoperability and collaboration demand data and information sharing, we need more open semantic-based applications that are able to interoperate and collaborate with each other. Ability of the system to perform semantically enhanced resource search/browsing based on resource semantic description brings a valuable benefit for today Web and for the Web of the future with unlimited amount of resources. Proposed resource visualization approach can find a place and can be utilized in various visual systems and especially in next-generation human-centric open environments for resource collaboration with enhanced semantic and context-based visual resource browsing. Presented 4i (FOR EYE) technology quite fits the demands of a new generation of integration systems. It is an ensemble of Platform Intelligent GUI Shell and visualization modules – MetaProviders that provide context-dependent representation view of resource data and integration on two levels. These are: information (data) integration of the resources to be visualized; and integration of resource representation views with a handy resource browsing in different dimensions. It can be considered as a new valuable extension of text-based Semantic MediaWiki to Context-based Visual Semantic MediaWiki. With the idea of the GUN we come to the environment where all the resources are semantically interoperable and have own semantic description – Resource Semantic Track. With the growing ubiquity of digital media content, ability to combine continuous media data with its own multimedia specific content description into the one source brings the idea of a true multimedia semantic web one step closer. 4i is a good basis for the different business, production, maintenance, healthcare, social process models creation and multimedia content management.

In the context of UBIWARE, 4i (FOR EYE) technology is a part of it. From one side, the technology is a base for Human-Resource adaptation and will be elaborated accordingly to the principles and vision of UBIWARE. The intelligence of this smart interface is a result of collaboration of multiple agents: the human's agent, the agents representing resources of interest (those to be monitored or/and controlled or requesting a human), and the agents of various visualization services – MetaProviders via an agent of Human-Resource Adapter. From the other side, accordingly to the 4i, specific interfaces (MetaProviders) will be developed to provide functionality for a human (Platform administrator) to configure functionality of a UBIWARE-based system.

Progress statuses of the industrial cases – Deliverable 1.2

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 three industrial cases, those of ABB, Fingrid and Metso Automation.

During the Year 1, with respect to all three cases the following tasks have been performed:

- Case analysis: identification of relevant industrial resources, their dependencies and interactions
- Connecting to relevant industrial resources: Development of appropriate resource adapters

Fingrid case:

During case analysis phase the following systems were considered:

- *Event History Database* (Oracle) in the office environment, to which data is automatically replicated from SCADA's event history database.
    - The focus is on R1-alarms, i.e. equipment alarms that require some maintenance actions to be performed.
    - The database also keeps events on when a maintenance worker entered or left a substation (billing is based on working time).
    - R3, R4, and R5 alarms, i.e. disturbances in the network, can also be considered.
    - Major R3, R4, and R5 alarms are also manually fed into Fingrid extranet (sähkömarkinnat => käyttöhäiriöt page)
- *Elnet system* (Oracle) that stores information about assets: towers, feeders, substations (i.e. the whole power network) + which maintenance service provider serves which working area.
    - *MapInfo* GIS is used for geographic representation of the power network.
- *Tosu system* (MS Access) that is used by the maintenance service providers to report to Fingrid the costs for the work performed.

- *The lightning info-service* – FMI provides for Fingrid data on all the lightning events in Finland. Additionally, FMI provides a Google Maps –based application, which geographically shows lightnings data combined with the data on the locations of Fingrid power lines.

The goal of industrial prototype for the first year is to implement Statistic analysis of the Event History data using UBIWARE platform integrated with two industrial resources: Event History DB and a Human. Pilot application delivers reports on how many R1 alarms were happening per month /year per working area.

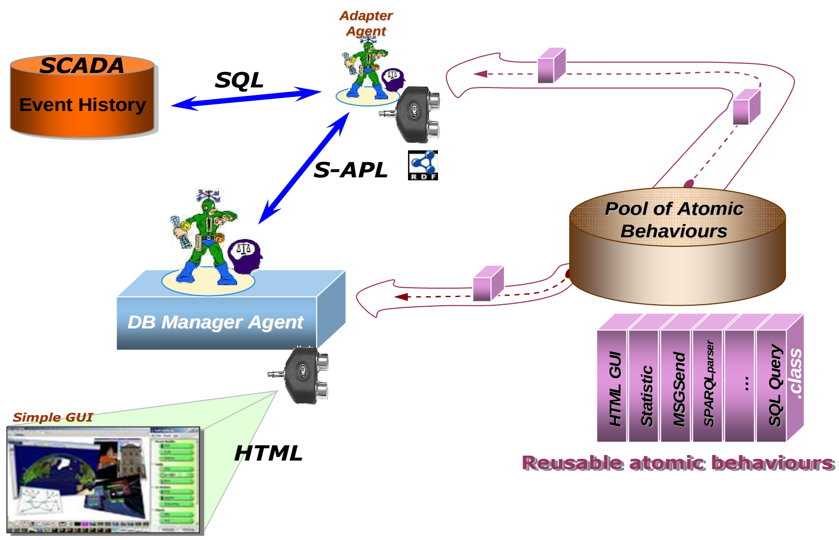


Figure 4 - Industrial prototype – Fingrid case

As future opportunities, according to case studies, the following functions could potentially be automated:

- *Extension of Statistic analysis of the Event History data*
  - Analysis of efficiency of maintenance service providers. In case of an R1 alarm in their working area, the provider is notified automatically. One question is how much time it takes the provider to reach the substation to perform maintenance.
  - Filtering out (as an option) events while “kuluvalvonta” is off
- *Integrating data from Event History, Elnet and Tosu*
  - Analysis of the relationship between alarms (Event history) and the types of equipment (Elnet).
  - Understanding what alarm has led to what maintenance actions at what cost.
- *Integrating data from Event History and Lightening data*
  - Matching the locations and times of R3,4,5 alarms with the locations and times of lightning strikes to automatically filter out lightning-caused disturbances (normally require no action to be performed).

ABB case:

During case analysis phase the following systems were considered:

- *Event History Table* (MicroSCADA). Event log that contains records about events on that happened on feeders, substation and control centre.

- *Weather station of KOILLIS-SATAKUNNAN SÄHKÖ OY(KSS).* Website (http://62.197.182.85/saa/index.html) contains history of weather parameters for last 8 days.

The goal of ABB industrial prototype for the first year is to provide Analysis of the Event History data along with simple classification for events based on descriptions provided by experts (Events annotation). This includes:

- Group events by feeder, substation.
- Analysis of automatic reclosing sequences and providing assistance to operator for event classification.

Prototype application consists of 2 agents: Event sender Agent and Control Agent. Event sender simulates appearing event in real time and pushes events to Control Agent. Control Agent receives events, does reasoning and produces HTML GUI.

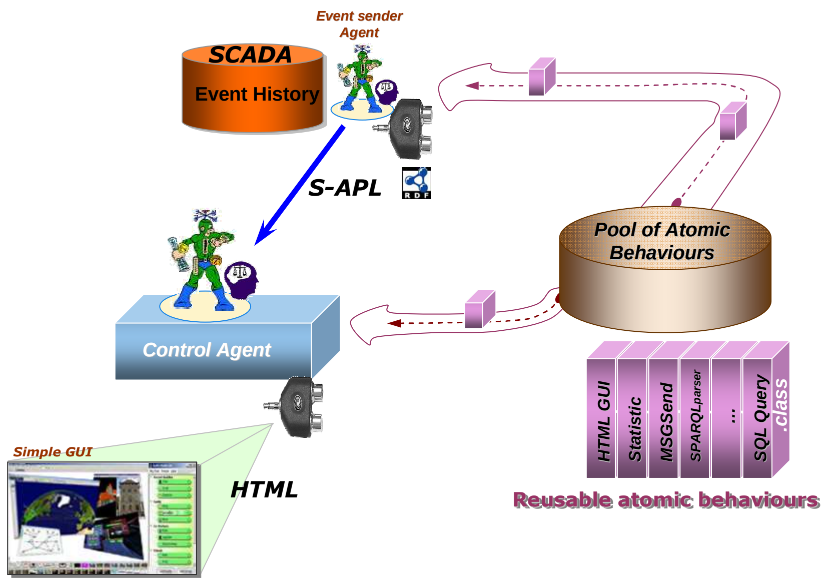


Figure 5 - Industrial prototype – ABB case

Future opportunities:

- *Integrating weather data.*
    - Analysis of the relationship between alarms in event history and the weather conditions.
- *Integrating feeders' location data* (feeders' coordinates)
    - Possibility to display feeders' locations on map.
- *Integrating information about landscape.*
    - Analysis of dependence different faults on landscape.

Metso case:

Metso Automation has been running research activities with Industrial Ontologies Group related to the Semantic Integration of industrial information for four years already. For the last one and a half years the cooperation strengthened and a separate privately funded project was launched. The project called SWIMMER aims at exploring the possibilities and add-values the Semantic

Technology can bring to enhance future products of Metso Automation. The Ubiware project has a specific case for Metso Automation, which complements the activities in the private project. Within the UBIWARE project Metso Automation expects to grasp UBIWARE platform functionality and capabilities. Special attention is paid to the adaptation of new sources and semantic querying of the industrial data. Lately there was an express of interest related to filtering and classification of events.

The prototype demonstrates the infrastructure for web service-based information flows to the UBIWARE platform, the adaptation of the received information and simple browsing of the collected data with primitive web-based interface, which issues queries to the platform via HTTP. Metso Automation extensively uses Web Service technology within the company that allows simple and fast creation of data flows to and from different sources. Metso Automation has arranged a flow of SOAP/XML messages to the University server. These messages may contain information from different customers and different information systems. Metso sends us only those, which they consider as interesting ones for information integration and future querying. The schema below (see Figure 6) describes the setup for storage of the information from Metso.

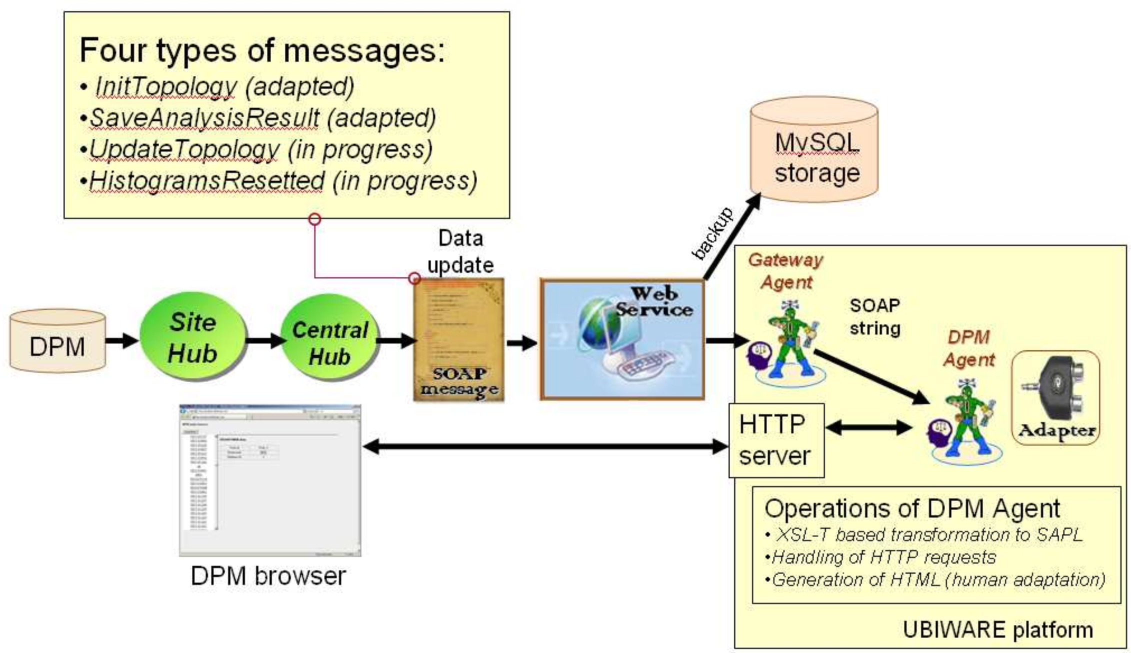


Figure 6 - Infrastructure for information flow from Metso

Future opportunities:

Amongst future opportunities considered by Metso Automation and planned as a cooperative future work with Industrial Ontologies Group is strengthening of the joint SW project involving experts from both sides to incorporate the semantic features into new Metso Automation's products.

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

During WP1's Year 1 (the *Representation* phase), the Semantic Agent Programming Language (S-APL) was developed which is a semantic tool sufficient for describing all of the following: resources' properties, agents' behaviors, organizational policies, business processes, ontologies, etc. Having a common language for all those ensures that any type of information in UBIWARE is semantic and facilitates intertwining those: e.g. one can easily prescribe that in a certain business process, if some properties of some resource have some certain values this must lead to some specific action taken by some agent unless prohibited by some organizational (e.g. security) policy.

The central to the core platform is the architecture of a UBIWARE agent depicted in the Figure 7. There is the behavior engine implemented in Java, a declarative middle layer, and a set of sensors and actuators which are again Java components. The latter we refer to as *Reusable Atomic Behaviors (RABs)*. We do not restrict RABs to be only sensors or actuators, i.e. components sensing or affecting the agent's environment. A RAB can also be a reasoner (data processor) if some of the logic needed is impossible or is not efficient to realize with the S-APL means, or if one wants to enable an agent to do some other kind of reasoning beyond the rule-based one.

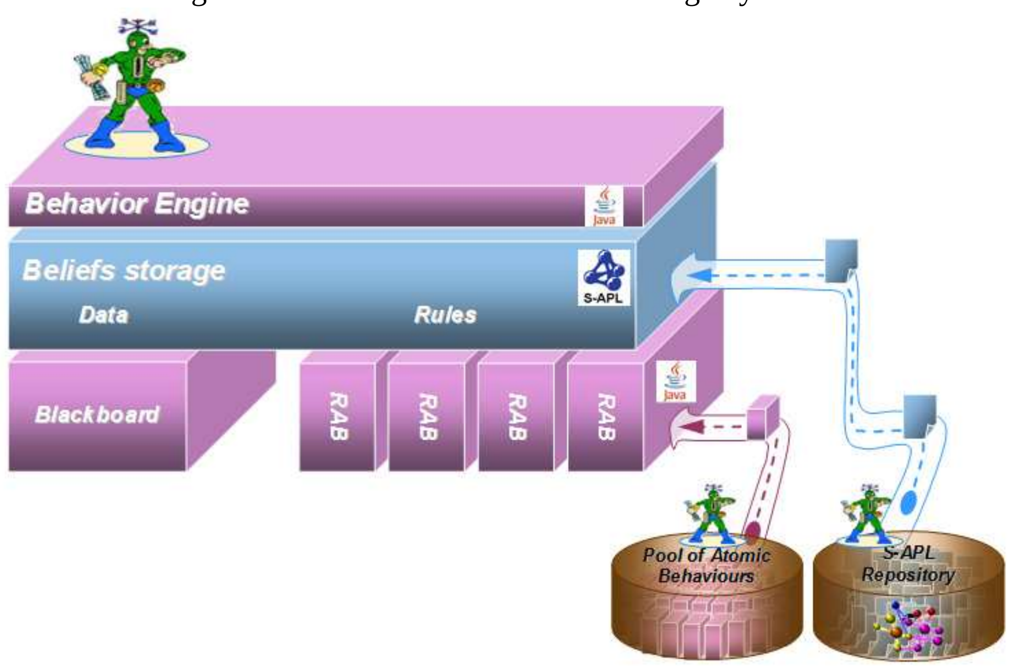


Figure 7 - The UBIWARE agent architecture

The UBIWARE agent architecture implies that a particular UBIWARE-based software application will consist of a set of S-APL documents (data and behavior models) and a set of specific atomic behaviors needed for this particular application. Since reusability is an important UBIWARE concern, it is reasonable that the UBIWARE platform provides some of those ready-made. Therefore, the UBIWARE platform as such can be seen as consisting of the following three elements:

- The behavior engine
- A set of "standard" S-APL models.
- A set of "standard" RABs.

Then, also the development outputs of other UBIWARE work packages are exactly some sets of such "standard" S-APL models and RABs that can be used by the developers to embed into their applications certain UBIWARE features such as security or flexible declarative (re-) configuration.

Also, we added an important additional element of the UBIWARE platform - a set of tools facilitating the development of UBIWARE-based applications. One tool in this group has already been under development and its first version is included with UBIWARE 1.0. It is the beliefs-visualizer / debugger interface. Any agent can start its own visualizer (it is done by using DebugBehavior RAB) to enable the developer to monitor the agent's beliefs and also to control the agent's execution.

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.

As to UBIWARE 1.0 platform, the development contribution of WP2 consists of:

- A set of standard S-APL models: Communication.Listener, Communication.Informer, Communication.Follower and Communication.Believer;
- A set of enhancements to the standard RABs MessageSenderBehavior and MessageReceiverBehavior which are used for message exchange between UBIWARE agents;
- Some modification in the behavior engine to enable implementation of the approach.

Use of S-APL as a communication language is quite natural because the communication over S-APL is easily organized with S-APL programming alone. An obvious additional benefit is the level of integration that in no-effort is then achieved between the communication and the agent behavior prescriptions. For example, an agent can query another agent for behavior rules – either to understand how that will react if a certain situation occurs or to learn itself how to achieve a certain goal. Similarly, agents can exchange commitments, plans, or basically belief structures of any complexity.

WP3: SURPAS

The security is often seen as an add-on feature of a system. However, in many systems (and UBIWARE is one of them), the system remains nothing more but a research prototype, without a real potential of practical use, until an adequate security infrastructure is embedded into it. The main objective of this work package is the design of the SURPAS infrastructure for policy-based optimal collecting, composing, configuring and provisioning of security measures in multi-agent systems like UBIWARE. SURPAS follows the general UBIWARE vision – configuring and adding new functionality to the underlying industrial environment on-the-fly by changing high level declarative descriptions. Regarding security, this means that SURPAS will be able of smoothly including new, and reconfiguring existing, security mechanisms, for the optimal and secure state of the UBIWARE-based system, in response to the dynamically changing environment. The optimal state is always a tradeoff between security and other qualities like performance, functionality, usability, applicability and other.

During WP3's Year 1 (the *Access* phase), WP3 developed an infrastructure for semantic access control in UBIWARE, with S-APL used for specification of access control policies. Such policies may prohibit or allow an operation of a certain class to be performed by an agent of a certain class on some resource of a certain class.

As to UBIWARE 1.0 platform, the development contribution of WP3 consists of:

- Introducing into S-APL and the behavior engine the concept of a meta-rule;
- Standard S-APL models Security.SBACReasoner and Communication.Ontology.

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 1 (the *Component* phase), we developed solutions for configurability of basic UBIWARE elements such as resource adapters and Reusable Atomic Behaviors, with S-APL used as the tool for both describing the configuration and for applying it.

As to UBIWARE 1.0 platform, the development contribution of WP4 consists of:

- A set of RABs - TextTableReaderBehavior, SQLReaderBehavior, ExcelReaderBehavior, XmlReaderBehavior, XmlWriterBehavior - as a part of the general approach towards resource adaptation;
- Standard S-APL models Configurability.RABConfigurator.

Roughly speaking, resource adaptation involves accessing data from an industrial resource (either physical through sensors or a digital like a database) in its own proprietary format and transforming this data into an ontological S-APL presentation. The UBIWARE's general approach towards configurable resource adaptation is depicted in Figure 8.

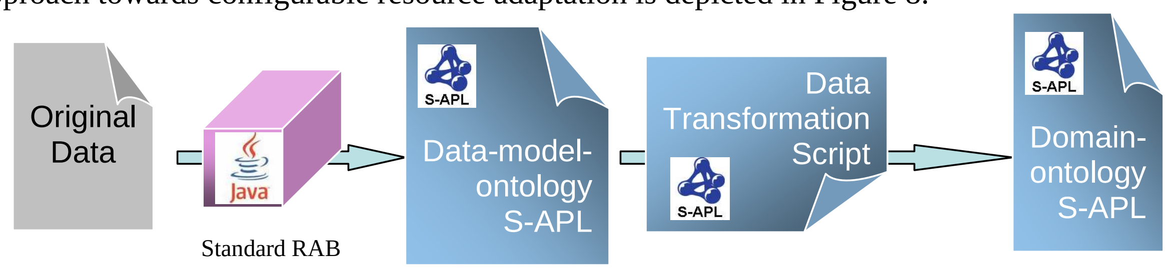


Figure 8 - UBIWARE approach to resource adaptation

The resource adaptation is performed in two phases:

- First, the original data from the resource is transformed into an S-APL representation based on the ontology of the data model of the original data. This transformation is performed by a Reusable Atomic Behavior (RAB).
- Second, the data from the data-model-ontology S-APL is transformed into the final domain-ontology S-APL using S-APL own means, i.e. rules.

At present, most of the data in the industrial applications is encoded using either *table data model* (relational databases, comma-separated files, etc) or *XML tree data model*. Ubiware 1.0 provides a set of standard transformation RABs, including TextTableReaderBehavior (file), SQLReaderBehavior (relational database), ExcelReaderBehavior (MS Excel table), and XmlReaderBehavior (XML file). These 4 RABs already cover a large share of industrial adaptation cases. Note UBIWARE 1.0 also includes XmlWriterBehavior which performs the transformation in the opposite direction. Obviously, the whole resource adaptation process in Figure 7 may be performed in the opposite direction - we transform some domain-ontology-based S-APL into XML-tree-based S-APL and then apply XmlWriterBehavior to get the final XML document. We use this approach in the industrial cases to produce HTML interfaces.

In UBIWARE, the meta-rule mechanism, which has been initially introduced for realizing security access control policies, is also used for configurability of Reusable Atomic Behaviors (RABs) (Actually the approach works for any unconditional commitments - also to complex actions not having “java:” or “:” namespaces, but we will speak here about RABs).

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.

During the Project Year 1, the work in this WP develops the general principles of the 4i approach and is aimed at developing an appropriate GUI Shell.
The WP development task *Task T1.2_w5* for the Year 1 is concentrated on development of a simple MetaProvider for context dependent resource visualization, development of initial GUI-Shell able to communicate with the MetaProvider.

SmartInterface is presented by GUI-Shell (central browser of the system) and remote distributed visualization modules - MetaProviders. GUI-Shell is presented by Html-page and remote server part that plays role of search engine and performs all necessary complex calculations (see Figure 9).

Concerning development of the simple MetaProvider (“Member-Of visualizer”) that visualizes resource in contexts were “member-of” property plays main role, with the purpose to visualize it in more natural for human way, we decided to utilize X3D[4] related technologies. X3D is a royalty-free open standards file format and run-time architecture to represent and communicate 3D scenes and objects using XML. It is an ISO ratified standard that provides a system for the storage, retrieval and playback of real time graphics content embedded in applications, all within an open architecture to support a wide array of domains and user scenarios. The development of real-time communication of 3D data across all applications and network applications has evolved from its beginnings as the Virtual Reality Modeling Language (VRML) to the considerably more mature and refined X3D standard. To provide communication with X3D scene, we utilized SAI (Scene Access Interface) that allows a programmer to change or build X3D worlds, and AJAX technology.

[4] X3D technology - http://www.web3d.org/

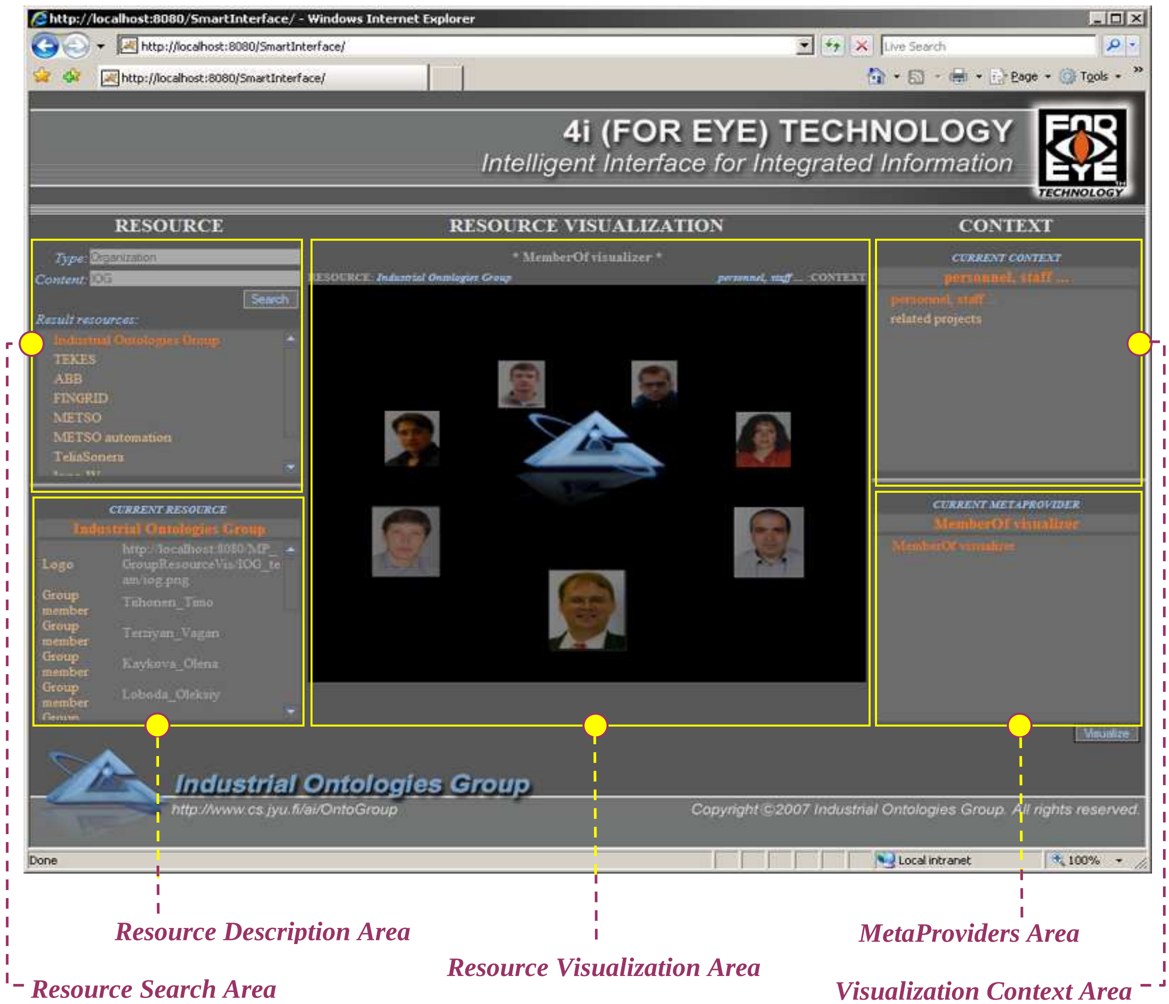


Figure 9 – GUI-Shell.

Following figure (see Figure 10) presents a full interaction model of the system.

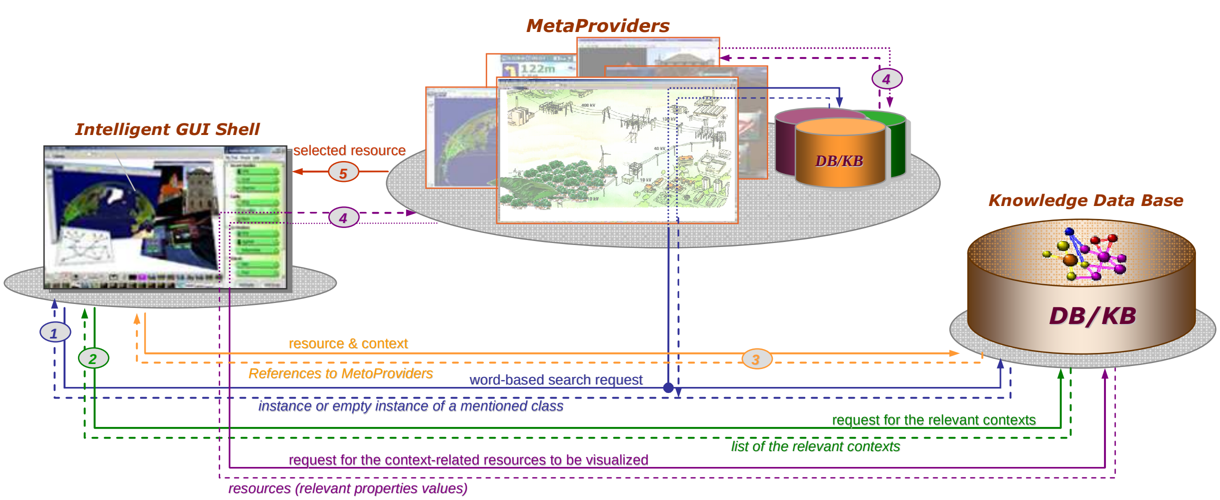


Figure 10 – SmartInterface interaction model.

4 Evaluation of the Project Results (Year 2007-2008)

Scientific Value

First of all, the project results gained during its first year (2007-2008) have quite a significant scientific value. The research efforts have been addressed to exploring the beneficial features of a synergy between young Semantic Web technology and more mature technology of Multi-Agent Systems. More precisely, we have concentrated on development Semantic Agent Programming Language (S-APL) - APL based on theW3C's Resource Description Framework (RDF). We presented our long-term vision for the security and privacy management in complex environments, SURPAS. It aims at policy-based optimal collecting, composing, configuring and provisioning of security measures in multi-agent systems like UBIWARE. We have paid attention on the configurability that allow us to change the process with the anticipation of possible changes and making them a part of the functionality in the very beginning, thus giving more flexibility to the business process management. At the same time, we do not forget about a role of a human in such new integration environments and studied dynamic context-aware Agent-to-Human interaction in UBIWARE, and elaborated a technology which we refer to as 4i (FOR EYE) technology.

The results of the research and development gained during 2007-2008 project year have been published in about 7(3) papers (see Section 6).

Partner Network Widening

Our main international collaborators:

- University of California, Berkeley (declarative networking);
- Massachusetts Institute of Technology, Data Center (semantics in RFID-based systems);
- University of Southern California (multi-agent systems, distributed constraints optimization, robots coordination in P2P environments);
- Lulea Technical University (smart services, embedded systems, telecommunications);
- VU Amsterdam (agents and Semantic Web);
- University of Athens (Service-Oriented Architectures);
- DERI, National University of Ireland, Galway (sensor networks middleware, Internet of things);
- University of Madeira (Semantic Web processes and services);
- and many others...

5 Further Development

The years 2 and 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.

WP1: UbiCore

During WP1's Year 2 (the *Deliberation* phase), we will work towards overcoming the following current limitation. Naturally, any specific agent has to load a set (more than one) of S-APL documents: own behavioral models, organizational policies, configuration settings, and other. So far, it was a responsibility of the developer to ensure that those S-APL documents do not conflict each other. Such absence of conflicts is not realistic. WP1 has to therefore answer 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 S-APL 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 WP tasks for the Year 2 are the following:

Task T2.1_w1 (research): Answers to the questions above, and other, if appear in the process, related to an agent's deliberation, i.e. acting based on the imposed behavior models. Design of the advanced behavior engine.

Task T2.2_w1 (development): Incorporating the research findings to the UBIWARE prototype.

WP2: UbiBlog

During WP2's Year 2 (the *Integration* phase), we will 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 WP tasks for the Year 2 are the following:

Task T2.1_w2 (research): Answer to the question above. Design of mechanisms for information querying and integration from distributed histories.

Task T2.2_w2 (development): Incorporating the research findings to the UBIWARE prototype.

WP3: SURPAS

According to the original project plan, the WP3's Year 2 (the *Communication* phase) had the following tasks:

*Task T2.1_w3 (research): C*onceptual and functional semantics, algorithms and abstract architecture: Secure Communication.

Task T2.2_w3 (development): Reference implementation of the abstract architecture in the UBIWARE prototype focusing on Secure Communication.

In other words, the plan was to work on the questions related to secure communication between agents, e.g. enabling organizational policies that do not only prohibit or allow something (like access control) but also may prescribe a certain additional actions to be taken, like encrypting every outgoing message. These questions are however already partially answered during first year, and will also be treated on the more general level of organizational policies of any kind in WP1. Because of that and because of reduced funding, it is considered to be reasonable not to perform work in this WP during Year 2.

WP4: COIN

During WP4's Year 2 (the *System* phase), we will 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)?

The WP tasks for the Year 2 are the following:

Task T2.1_w4 (research): Answer to the question above. Design of mechanisms for (re-)configuration of complex system in a distributed fashion.

Task T2.2_w4 (development): Incorporating the research findings to the UBIWARE prototype.

WP5: 4I (FOR EYE) technology

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?

The WP tasks for the Year 2 are the following:

Task T2.1_w5 (research): Answer to the questions above. Design of mechanisms for context-aware (visual) representations creation and combining.

Task T2.2_w5 (development): Incorporating the research findings to the UBIWARE prototype.

WP6: Middleware for Peer-to-Peer Discovery

The WP tasks for the Year 2 (The *Discovery* phase) are the following:

Task T2.1_w6 (research): Design of a P2P resource discovery infrastructure for UBIWARE, including related inter-agent communication protocols.

Task T2.2_w6 (development): Incorporating the research findings to the UBIWARE prototype.

WP7: Industrial Cases and Prototypes

There are four industrial cases, those of ABB, Fingrid, Metso Automation and Nokia.

During the Year 1, with respect to the three former cases the tasks were related to case analysis, i.e. identification of relevant industrial resources, their dependencies and interactions, connecting to some of those resources and realizing some simple interactions between them. The Nokia case is a new one, introduced in Year 2.

During the Year 2, with respect to all four cases the task is the following:

Task T2.1_w7: Developing a full prototype application: connecting to additional relevant resources and extending the interactions between them towards a sufficiently elaborated application.

6 UBIWARE Project Publications (up to the April of year 2008)*

[1] 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, 10 pp. (to appear).

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

[3] 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

[4] 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.

[5] 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.

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

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

Submitted or going to be submitted:

[8] Katasonov A., Terziyan, V., Semantic Approach to Engineering Multi-Agent Systems, In: Proceedings of the 1st international workshop on Agents for Autonomic Computing (AAC 2008) in

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

conjunction with the 5th International Conference on Autonomic Computing (ICAC 2008). June 6, 2008, Chicago, USA, IEEE CS Press, 8 pp. (submitted 21 April 2008).

[9] Katasonov A., Terziyan, V., Agent Communications with S-APL as the Content Language, In: Proceedings of the International Workshop on Middleware for the Semantic Web in conjunction with the Second IEEE International Conference on Semantic Computing (ICSC-2008), August 4-7, 2008, Santa Clara, CA, USA, IEEE CS Press, 6 pp. (submitted 22 April 2008).

[10] Khriyenko O., SMART HUMAN ADAPTER - Multi-agent Context-sensitive Visual Resource Browser, In: Proceedings of the Twelfth International Workshop on Cooperative Information Agents (CIA-2008), September 10-12, 2008, Prague, Czech Republic, Springer LNCS, 15 pp. (submitted 21 April 2008).

7 References

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

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

Nixon, L., (2006). Multimedia, Web 2.0 and the Semantic Web: a strategy for synergy. In: First International Workshop on Semantic Web Annotations for Multimedia (SWAMM), as a part of the 15th World Wide Web Conference, Edinburgh, Scotland, 22-26 May, 2006.

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.

Yuxin, M., Zhaohui, W., Zhao, X., Huajun, C., Yumeng, Y., (2005). Interactive Semantic-Based Visualization Environment for Traditional Chinese Medicine Information. In: Web Technologies Research and Development - APWeb 2005, 7th Asia-Pacific Web Conference, Shanghai, China, March 29 - April 1, 2005, Springer, volume 3399/2005, ISBN 978-3-540-25207-8, pp. 950-959.