TEKES

UBIWARE Project

Year 2007-2010

UBIWARE TEKES PROJECT – FINAL PROJECT REPORT (2010)

Final Project Report

UBIWARE: "Smart Semantic Middleware for Ubiquitous Computing"

Industrial Ontologies Group

Agora Center, University of Jyväskylä

University of Jyväskylä

metso automation FINGRID inno-w ECURAS ABB NOKIA Connecting People metso shared services

December, 2010

Jyväskylä, Finland

Main Contact Persons

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

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

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

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

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

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

Tekes Contact Person (2007-2009), technology expert **Timo Taskinen**,
TEKES.

timo.taskinen@tekes.fi , Tel.: 010 60 24846

Tekes Contact Person (2009-2010), technology expert **Katja Ahola**,
TEKES.

katja.ahola@tekes.fi , Tel.: 010 60 55815

Project's webpage:

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

Metso Automation

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

(2007 – 2010)

Fingrid

Fingrid has a crucial role in the Finnish power system. By virtue of the Electricity Market Act, the company is responsible for the functioning of the power system at a national level. This is vital in view of the welfare and development of the entire society and economy.

(2007 – 2010)

Inno-W

Inno-W provides collaborative and networked business solutions, based on Web 2.0 and Microsoft SharePoint technologies. Their special know-how is in corporate innovation management and networked communities.

(2007 – 2010)

Hansa Ecuras

Company specializes in audio, video and conferencing systems design, installation and maintenance. They utilize the latest digital technologies to design systems with respect to the installation site architecture and layout. Company also provides noise and acoustic measurements and planning.

(2007 – 2010)

ABB

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

(2007 – 2008)

Nokia

Nokia is the world leader in mobility, driving the transformation and growth of the converging Internet and communications industries. Company becomes the leading provider of mobile solutions. Its solutions strategy leverages one of its greatest assets - a portfolio of outstanding devices, with unmatched scale and geographic reach. Nokia couple them with smart services, integrated via an intuitive and seamless user experience. Company differentiates these solutions offerings based on its in-depth consumer understanding, with a strong focus on social location (people and places).

(2008 - 2009)

Research and Development Group

University of Jyväskylä, Agora Center

Industrial Ontologies Group

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

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

2007-2010

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

2007-2010

Oleksiy Khriyenko
Ph.D., Researcher & Developer
oleksiy.khriyenko@jyu.fi
http://www.cc.jyu.fi/~olkhriye/

2007-2010

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

2007-2010

Artem Katasonov
Ph.D., Researcher & Developer

2007-2009

Michal Nagy
M.Sc., Researcher & Developer
michal.nagy@jyu.fi

2008-2010

Michael Cochez
B.Sc., Researcher & Developer
michaelcochez@gmail.com

2008-2010

Anton Naumenko
Ph.D., Researcher & Developer

2007-2008

Joonas Kesäniemi
B.Sc., Trainee
joonas.h.v.kesaniemi@jyu.fi

2008-2010

Viljo Pilli-Sihvola
B.Sc., Trainee
viljo.pilli-sihvola@jyu.fi

2010

Jose Luis Garduno
B.Sc., Trainee
deepsight@gmail.com

2010

Arnim Bleier
M.Sc., Trainee

2008

Nikos Mouchtaris
B.Sc., Trainee

2008

Michal Szydlowski
B.Sc., Trainee

2008

Dmytro Zhovtobryukh
Ph.D., Researcher

2009

Table of Contents

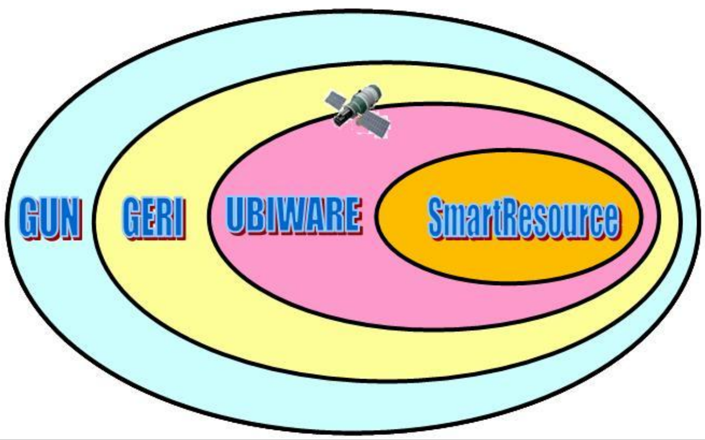
UBIWARE

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 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 (Figure3):

- 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.

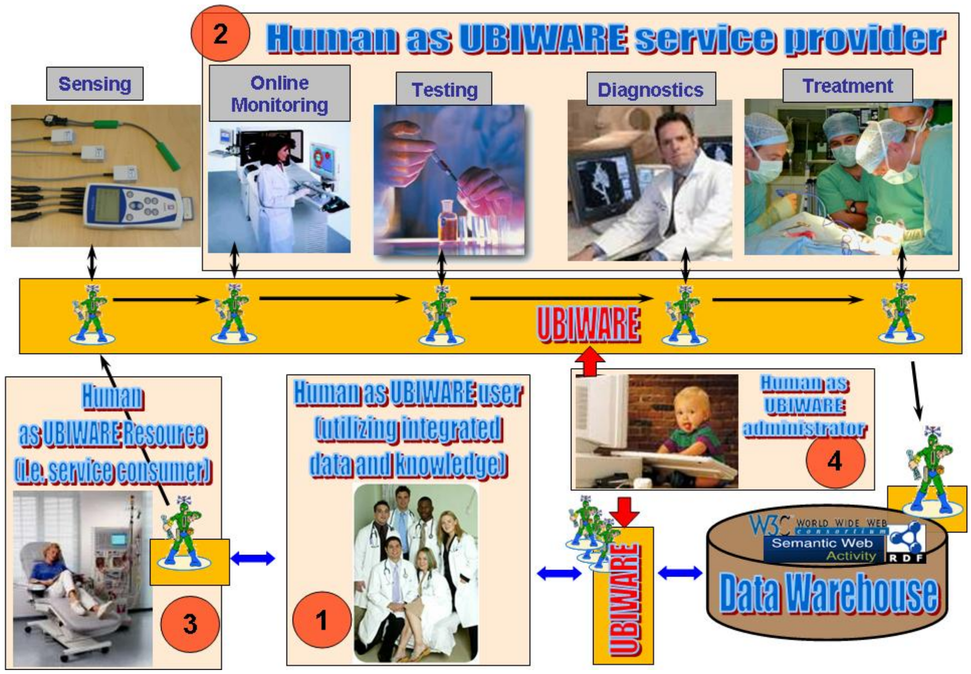


Figure 3 - Human in different roles in UBIWARE.

UBIWARE

Motivation and Goals

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” 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 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.

Self-management of systems is one of the central themes in the EU 7-th Framework ICT Program (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.

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

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.

UBIWARE

Project Stages

Work in this project is divided into six research and development workpackages which are running in parallel and complemented with WP7 that contains the industrial pilot cases and prototypes:

1. *WP1: Core Distributed AI platform design (UbiCore)*
2. *WP2: Managing Distributed Resource Histories (UbiBlog)*
3. *WP3: Smart Ubiquitous Resource Privacy and Security (SURPAS)*
4. *WP4: Self-Management, Configurability and Integration (COIN)*
5. *WP5: Smart Interfaces: Context-aware GUI for Integrated Data - 4I(FOR EYE) technology*
6. *WP6: Middleware for Peer-to-Peer Discovery (MP2P)*
7. *WP7: Industrial Cases and Prototypes.*

Due to lack of funding during the second project year, WP3 had been postponed. Later on, when the funding level was fixed during the last project year, it became possible to restart WP3. Thanks to development achieved in other WPs during years 1 and 2 we expand the scope of WP3 and rename it as WP3: *Policy Management Engine in UBIWARE*. Likewise the scope of WP6 is extended from P2P to other approaches in global integration and the WP6 is renamed as WP6: *MetaUBIWARE: Global Enterprise Resource Integration*.

Workpackages 1 through 6 include both research and development tasks. The tasks are approached by combining various research methods with agile software development processes. This means that software prototypes were iteratively developed during the whole project lifecycle based on real data, real needs and changing requirements of industrial partners. The theoretical part was presented in three separate deliverables (one per year) D1.1, D2.1, and D3.1. The practical results are 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, were developed during each project year, as UBIWARE 1.0, UBIWARE 2.0, UBIWARE 3.0 and UBIWARE 3.1 (due to prolongation of the project), and reported through deliverables D1.3, D2.3, D3.3 and D3.4, correspondingly. The status reports for the industrial cases are collected in separate deliverables (one per year) D1.2, D2.2, and D3.2.

WP1: Core Distributed AI platform design (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, that may 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.

WP2: Managing Distributed Resource Histories (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, therefore, naturally includes 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.

WP3: Policy Management Engine in UBIWARE (SURPAS)

The main objective of this workpackage 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.

Instead of only security policies we take into consideration general policy-based management of UBIWARE scenarios. The research phase conceptualizes the aspects of policy creation and annotation in S-APL; policy enforcement engine; policy-driven MAS scenario management; policy-driven (re)configuration etc. Prototype of policy enforcement engine and required S-APL constructs will be implemented to the UBIWARE platform.

Policy in UBIWARE is a general term to describe in what kind of state a group of agents; an organization and thus by extension the whole system is allowed to be. A closely related concept are Preferences, they tell that if the system is able to make a choice, that it should favor a certain option. For example if an agent has to buy a certain product, it should always prefer to buy the cheaper solution if it doesn't take much more time to deliver.

In a concrete UBIWARE scenario, one might have certain conditions to which the MAS must comply. An example of a Policy could be that if an agent senses a certain problem, it must (before doing anything else) start an alarm procedure. One important set of policies are the security policies. This package has thus become broader as it was previously defined. Probably the way security policies are defined is going to change in order to fit into the broader view of a policy.

Policies have a higher priority than the preferences in this working package since a Preference could be seen as a policy of taking one option instead of another one, it is thus a special case of a policy. In what follows, when talking about Policies, we're also talking about Preferences.

WP4: Self-Management, Configurability and Integration (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 workpackage aims at introducing configurability as a pervasive characteristic of UBIWARE and developing the technology which will systemize and formalize this feature of the platform.

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

- o Initial self-configuration
- o 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.

WP5: Context-aware Smart Interfaces for Integrated Data (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 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.

WP6: MetaUBIWARE: Global Enterprise Resource Integration (MP2P)

The objective of this workpackage is the design of mechanisms which will extend the scale of semantic resource discovery in UBIWARE with peer-to-peer discovery. Such mechanisms have to enable an agent:

- To discover agents playing a certain organizational role,
- To discover an agent (or agents) possessing certain needed information.
- To discover resources (through its agents) of certain type or possessing certain properties (e.g. a device in state X, or a web-resource providing some information searched for).

In all cases, existence of a central Directory Facilitator is not assumed, so the discovery is to be performed in peer-to-peer manner. This has at least two goals:

- Improving the survivability of UBIWARE-based systems. P2P is a complementary mechanism which can be utilized in an exception situation where the central Directory Facilitator became for some reason unavailable.
- Discovery across UBIWARE-based systems. There could be several UBIWARE-based agent platforms (applications) started up independently, each having own Directory Facilitator. While JADE (underlying agent framework of UBIWARE) supports communication among agents residing on different platforms, it does not provide for the cross-platform discovery.

This workpackage studies and incorporates observation-based interaction to the Ubiware platform. Since observation is relatively new concept in the area of multi-agent systems, some of the research effort was concentrated on further refining the fundamental idea behind observation in the context of Ubiware and its benefits related to different facets of agent-based systems. The research involved comparison of communication and observation based interaction on both conceptual and practical level, including running use cases on the prototype platform.

We have considered the observation as a capability and, hence, applied adjusted Ontonuts technology (see deliverable D2.1) for capability sharing and consumption. The ontonut-driven observation as a service of Agent Observable Environment (AOE) can be smoothly incorporated in S-APL. For further development of the AOE concept we apply a new level of agents to the Ubiware, so-called platform agents. Conceptually platform agents can be seen as residing between application agents and the execution platform running the multi-agent system. Whereas the agents at the application level are designed to run business logic, the intentions and goals of the platform agents are geared towards more infrastructural crosscutting functions like scalability, security and availability. WP6 focuses on implementing AOE using platform level agents. Similar approach could be used to provide, for example, automatic replication and relocation services for application agents. Due to special role of platform agents, it might be necessary to optimize their implementation, for example, by using more efficient communication protocol than ACL in JADE.

When the notion of AOE is extended to the inter-platform level (GERI), the Ubiware platforms may cooperate via platform level agents, thus, allowing platforms to be represented at each other's site as platform agents with their globally shared capabilities, desires, intentions and actions. An important issue related to AOE, is to design and produce ontology for modeling the Agent Observable Environment. The AOE ontology will establish the basis for observation-based coordination inside and between Ubiware platform(s) as well as spontaneous interaction and observation based learning.

WP7: Industrial Cases and Prototypes

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.

UBIWARE

Project Results

Theoretical achievements

One of the main theoretical results of the UBIWARE is *Semantic Agent Programming Language (S-APL)*, which is a Resource Description Framework (RDF) - based language integrating features of several kinds of tools: agent programming languages (like AgentSpeak and AFAPL), semantic reasoners (like CWM), querying languages (like SPARQL) and agent communication content languages (like FIPA SL).

Semantic APL (S-APL) has been designed to meet the following requirements:

- 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 modeled.
- 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. Therefore, there is a basis for better understanding among agents with a goal of better coordination and collaboration among them.

S-APL enable not just traditional choreography and orchestration but much more powerful ones: *configurable and semantic choreography and orchestration.* This makes qualitative difference between S-APL and e.g. *BPEL* (*Business Process Execution Language* – very popular industrial standard). With S-APL one may describe logic of a *configurable process* (structure and components of the process are not explicitly prescribed but described in terms of intended properties and can be changed even during the process execution). In the same time with S-APL one may describe logic of *semantic integration* of data taken from external sources and reasoning logic on top of the data.

The S-APL engine was implemented and is central component of UBIWARE Platform and provides a base that is being used now for realizing all other UBIWARE functionality. Detailed information about S-APL can be found in the deliverable D1.1 and related publications (see Publication 17, 9)

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. 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. 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. In our approach (see deliverable D1.1) we designed 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. We have used 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 are the symmetry and expressive power in the UBIWARE platform are maximized. The agents are 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; or query if the other agent knows a rule that should be applied in a particular situation.

We have introduced Ontonut technology to tackle the problem of distributed querying in UBIWARE-based multi-agent systems (see deliverable D2.1 and Publication 12). 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 Ontonut 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. 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. 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. Ontonut concept provides descriptive mechanisms for agent components. The true dynamism can be reached, when these components can be advertised to other agents (e.g. in a form of services). Furthermore, agents should be able to search for advertised service components and negotiate with other agents about conditions of service consumption. We organized such functionality following the Ontonut approach to allow platform agents to plan, run and dynamically manage their processes with higher degree of freedom (see deliverable D3.1). Generalizing all above, Ontonuts technology 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. Due to *ontonuts* technology and based on the *Open World Assumption* (alternatively to BPEL with the Closed World Assumption), S-APL engine is able to discover and automatically utilize (in a runtime) new services (capabilities) that have just appeared in semantic service registry.

As a proof of concept, we demonstrated the data mining capabilities as services. The research presented in deliverable D3.1 describes specific domain of data mining services. We foresee that model player services will be a successful business case for the emerging paradigm of cloud computing. Pay-per-use principles combined with high computational capacities of cloud and standardized DM-models will be definitely an alternative to expensive business intelligence and statistics toolkits. Another niche of data mining services in cloud computing can be model construction services. Such systems will drive innovations in data mining methods as well as applied data mining in certain domains. Such service will compete by introducing know-how and innovative tools and algorithms that bring add-values in e.g. predictive diagnostics or computational error estimation. This direction will lead to so-called “web of intelligence”. The role of UBIWARE in cloud computing emerges as a cross-cutting management and configuration glue for interoperability of future intelligent cloud services. The main burden of UBIWARE will be management of consistency across different domain conceptualizations (Ontologies) and cross-domain middleware components. Fine-grained ontology modeling is still a challenge for research community and we predict that in the nearest future the domain modeling will be task-driven, i.e. the domain model engineers may incorporate some standardized and accepted conceptualizations, whereas the whole ontology for solution will be tailor made. Tailored ontologies will require subsequent mapping mechanisms and additional efforts. Nevertheless, we have to cope with it because building one centralized world ontology has been reasonably criticized as utopia.

In this project we also 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. 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. In the deliverable D2.1 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. 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. We proposed architecture for initial configuration involving Abstract process repository, director agent, ODF and several Ontonut implementations. Deliverable D3.1 described the initial process configuration phase, in which an executable process is built based on an abstract process.

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 (Smart Ubiquitous Resource Privacy and Security). It aims at policy-based optimal collecting, composing, configuring and provisioning of security measures in multi-agent systems like UBIWARE. 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 (see deliverable D1.1). The questions, related to policies, privacy and security also have been treated on the more general level of organizational policies of any kind (see deliverable D3.1 and Publication 5).

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. We proposed an approach of semantically enhanced context-dependent multidimensional resource visualization that 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. This approach is meant to be used 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 (see deliverable D1.1, Publication 18, 16, 19 and 20). Elaborated 4I(FOR EYE) technology is a new paradigm of GUI development. It enables creation of context-sensitive 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.

One of the most important features of open collaborative and dynamic environments is a so called “context awareness and 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. Following this approach, we elaborated context-aware Distance Measurement Function that measures a similarity between resources taking into account values of resource properties and weights of their significances (see deliverable D2.1 and Publication 11). We included a visualization of the resources in a context of their similarity/closeness to the 4I(FOR EYE) Browser as its an inherent functionality that allows to perform a search of similar resources. Also we have presented intelligent way of automatic/semiautomatic context recognition and personalized visualization invocation as a next valuable enhancement of the Browser. That was a first draft of the idea and it should be elaborated more comprehensive in the future (see deliverable D3.1).

Finally, to develop a solid functional browser that follows provided approach, that contains all mentioned features and make it a browser of new generation, we have elaborated architecture of 4I Browser with a flexible modular systems of source data adaptation, visualization context definition and browser extension with a new MetaProvider (see deliverable D3.1) . At the same time, concerning the commercialization steps of the Browser, we have considered general model of 4I Environment where we tried to define main players and roles. We defined two business scenarios of Browser utilization: *Global Use of the Browser* and *Local Corporate Use of the Browser*.

With respect to Global Enterprise Resource Integration and Peer-to-Peer Discovery issues, we have presented three different approaches of building distributed peer-to-peer infrastructure in multiplatform environments. We have also shown required modifications and additions that have to be incorporated into platform's components in order to employ presented mechanism. We described the instruments of active improvements in topologies of created networks. By the means of inter-platform discovery we give agents the opportunity to communicate, share services and resources beyond the boundaries of their home platforms.

UBIWARE Platform

The basic approach in development has been that of agile development – creation of a succession of prototypes with improving functionalities on every release combined with concrete use cases with companies.

The core of the UBIWARE platform gives 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 enables the resources to automatically discover each other and to configure a system with complex functionality based on the atomic functionalities of the resources. It ensures 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 UBIWARE Platform is a development framework for creating multi-agent systems. It is built on the top of the Java Agent Development Framework (JADE[2]), which is a Java implementation of IEEE FIPA specifications. JADE provides communication infrastructure, agent lifecycle management, agent directory-based discovery and other standard services. In UBIWARE project, a multi-agent system is seen, first of all, as a middleware providing interoperability of heterogeneous (industrial) resources and making them proactive and in a way smart. The central to the UBIWARE Platform is the architecture of a UBIWARE agent depicted in Figure 4. It can be seen as consisting of three layers: the Behavior Engine implemented in Java, a declarative middle-layer (Behavior Models corresponding to different roles the agent plays), 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 concerned with 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 S-APL, or if one wants to enable an agent to do some other kind of reasoning beyond the rule-based one. Current version of UBIWARE platform contains a set of Reusable Atomic Behaviors (RABs) and the libraries that simplify UBIWARE application development.

The middle layer is the beliefs storage. What differentiates S-APL from traditional APLs is that S-APL is RDF-based. This provides the advantages of the semantic data model and reasoning. An additional

[2] JADE - http://jade.tilab.com/

advantage is that in SAPL the difference between the data and the program code is only logical but not any principal. Data and code use the same storage, not two separate ones. This also means that: a rule upon its execution can add or remove another rule, the existence or absence of a rule can be used as a premise of another rule, and so on. None of these is normally possible in traditional APLs treating rules as special data structures principally different from normal beliefs which are n-ary predicates. S-APL is very symmetric with respect to this – anything that can be done to a simple statement can also be done to any belief structure of any complexity.

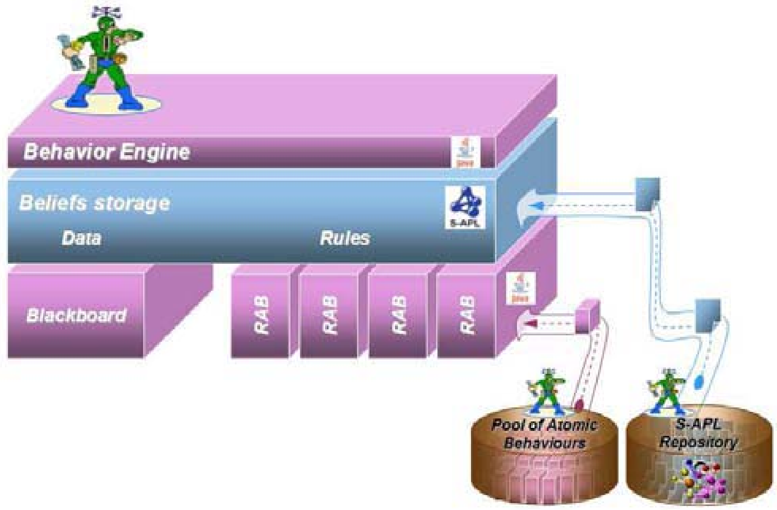


Figure 4 – UBIWARE Agent architecture.

Together with the main UBIWARE engine, we have implemented Ontonuts engine. 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.

Later, we have brought the UBIWARE platform to the qualitatively new level of the middleware solution – the platform now combines the features of the application server, the semantic web platform and the agent-driven platform, where agent-driven semantic applications can serve end customers with the high quality web-based GUIs, enhanced user-friendliness and responsiveness. The platform has become an application-independent runtime environment, where special infrastructure agents take care of the platform itself, not of the applications being run on it. At the same time, we introduce personal user agents, thus making the platform user-oriented.

We design the UBIWARE platform infrastructure for creation of various kinds of applications. Those applications have a freedom to use a web front-end, on-the-platform user management and other infrastructure or define their own platform components depending on the needs of the application. New architecture of the platform follows cloud computing paradigm. We identify two groups of agents. The first group includes the agents which are application-specific, whereas the second group gathers infrastructure agents providing services to those application-specific ones.

In the last version of the platform, we bring a new agent classification system that simplifies the management and policy control of agents. The new agent classification enabled new platform architecture. Now, the platform contains seven platform infrastructure agents, each supporting one type of platform functionality. At the same time we enhanced the platform with new packaging system for UBIWARE applications.

As a part of the Platform, we developed two new web applications supporting the platform administration. First application is used for user management and the second one is used for management of UBIWARE applications.

The UBIWARE platform itself has been now used to run several test cases, extended in various ways with new capabilities based on the research findings made in the project. Moreover, it has been tested with students as a tool for creating distributed autonomous systems as part of the course work. However, the platform has not been augmented so far with user interface tools etc. that would better support its use in software engineering. Hence the platform is still a prototype, not really a software product that could be exploited easily outside the project group.

Team has a clear vision on further improvement and enhancement of the platform. UBIWARE 4.0 will be finished after the project. It will include scenario editor, better user-as-developer interface, and payment system as an internal reusable functionality and will be open for updates.

Together with the development of UBIWARE platform, we developed a prototype of 4I Browser to proof the concepts behind proposed 4I (FOR EYE) technology: Intelligent Interface for Integrated Information. 4I (FOR EYE) is a smart ensemble of Intelligent GUI-Shell (smart middleware for context dependent use and combination of a variety of different MetaProviders, depending on the user needs) and MetaProviders as graphical interfaces that visualize filtered integrated information.

4I Browser is a mashuper, kind of engine that provides context-sensitive visualization of resources via MetaProviders, it provides interoperability between different resources and services and adds some additional functionality. Browser has an intelligent technique for automatic dynamic selection of a visualization context. 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.

Following the idea of search system we decided to include a visualization of the resources in a context of their similarity/closeness to the 4I Browser as its an inherent functionality. The main development in this direction has been done during the Inno-W industrial case development and includes elaboration of Similarity Distance Measuring Function and development of resource closeness/similarity Visualization component. Additionally, we extend Browser functionality with a context editor. Current implementation of the 4I GUI Shell supports visual configuration of resource similarity visualization context.

To make Browser able to work with any external repository, we have elaborated general adapter that enable to convert data from any format to the internal one. New data formats appear all the time and will require new adaptation modules. Thus, we decided to make this adaptation module extendible, be able to add new adaptation sub-module for new data format transformation. With a purpose to allow user to import any repository (in RDF or N-tuple format) him(her)self, we elaborated general adapter. Adapter is supplied with a graphical interface that helps user to configure adapter with a respect to supported data fields of internal format to be browsed through 4I Browser in the context of close/similar resources.

Development related to 4I (FOR EYE) technology has been started from early beginning of the project. Due to this fact, development of 4I Browser is done separately from the UBIWARE Platform, development of which has been started in the second part of the first project year. Nevertheless, development has been done following the common paradigm and approach. Due to component-based structure of the Browser and visualization modules it is possible to base the Browser on UBIWARE Platform without huge efforts.

Case Studies

Another practical results of the project, additionally to UBIWARE Platform related development, are case studies, applications that we have built based on the Platform. Some of them belong to Companies' needs; others are developed as a proof of concepts behind a certain functionality of the Platform.

Regarding to the case studies, we have two major goals. 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.

During the years of the project we have been developing five industrial cases for Fingrid, Metso Automation, Inno-W, ABB and Nokia Companies.

Fingrid industrial prototype

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:

- o *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.
- o *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.

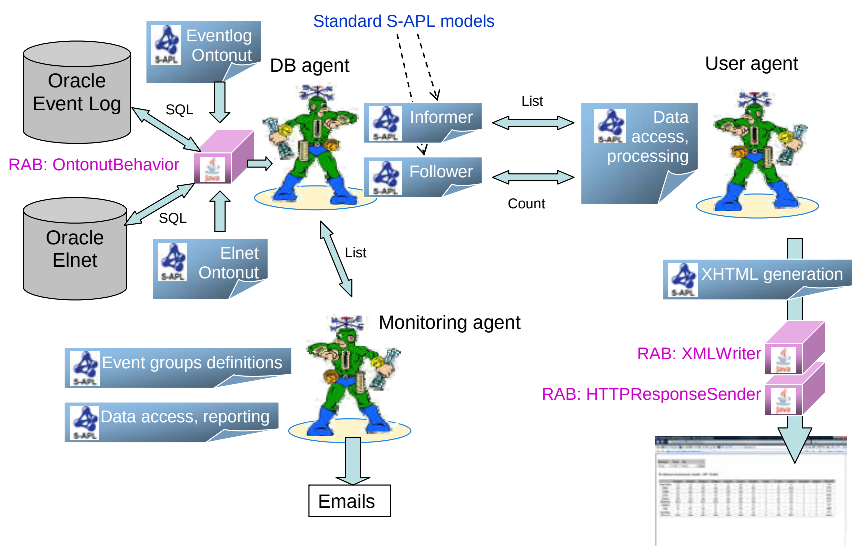


Figure 5 – Architecture of Fingrid prototype.

The goal of industrial prototype for the first year was 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.

During the second project year, *Data security* became a central concern in the case. It is major reason, along with *safety*, for putting the focus on historic analysis of events rather on the real-time operation. In result, the UBIWARE project team did not have a direct access to Fingrid databases. Rather, a surrogate Oracle database had to be created and populated with data provided by Fingrid. This database has two tables imitating the schemes of Eventlog and Elnet, and was sufficient for the application development purposes.

Later, the prototype was connected to the real databases in Fingrid office environment. Some inconsistencies between the real and the surrogate databases appeared, but were successfully resolved.

Metso Automation industrial prototype

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.

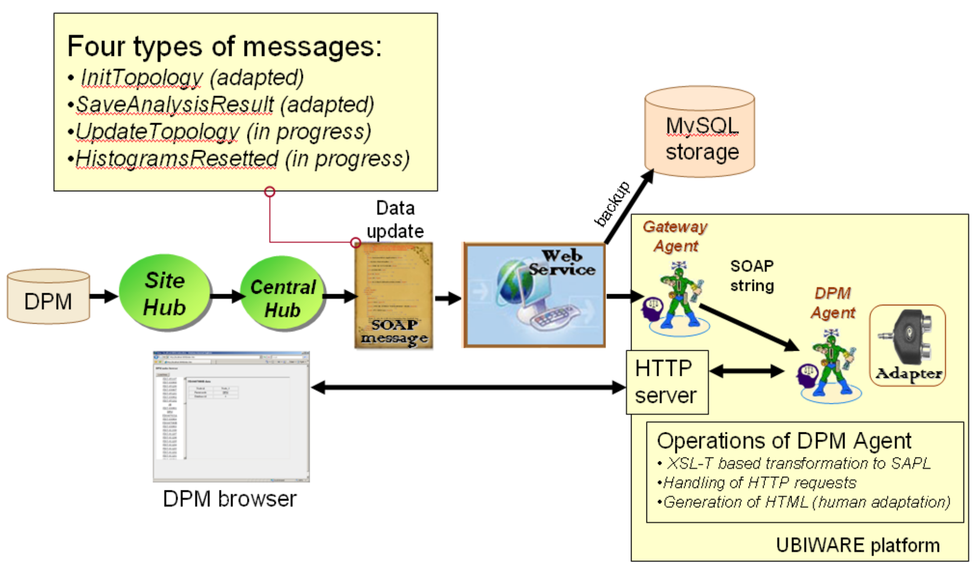


Figure 6 - Infrastructure for information flow from Metso.

Latter, 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 cannot 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:

- o Alarm messages collected in RDF format (messages have been collected during two years)
- o A sample of the DPM database (performance of each node of the paper machine)
- o A sample of the Diary database (events handled and documented by factory workers)
- o 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 description/annotation of the components refers to the domain ontology, which makes the specification explicit. At the same time, the descriptions themselves are not enough to run queries; they are interpreted by the Ontonuts engine and then used in planning and execution.

In the last version of the prototype we have made a manager for the components/Ontonuts developed. The manager provides functionality for editing and making test runs for the newly updated component descriptions.

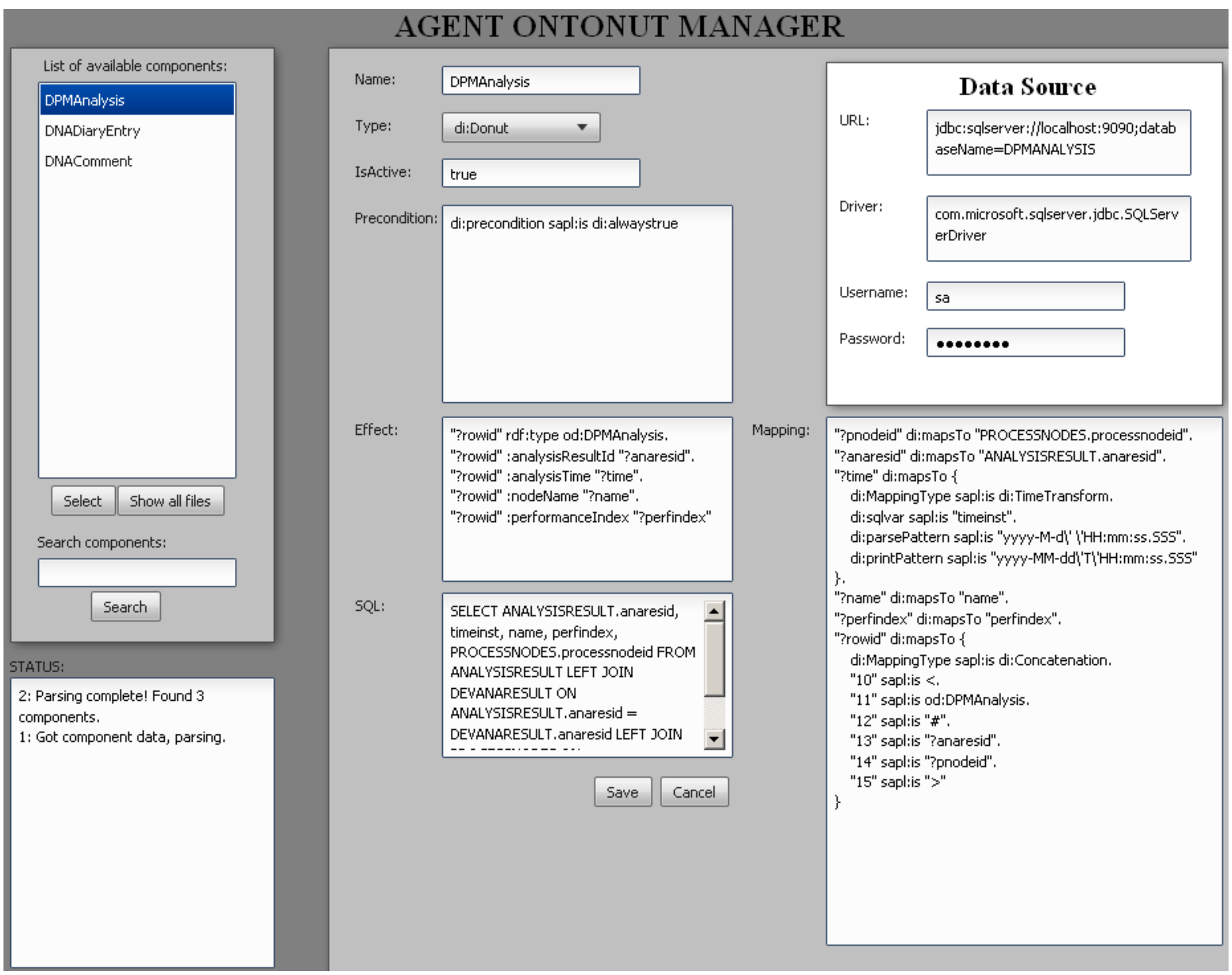


Figure 7 – Ontonut Manager.

Inno-W industrial prototype

The goal of Inno-W 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.

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 (based on elaborated Similarity Distance Measuring Function) is performed before visualization phase.

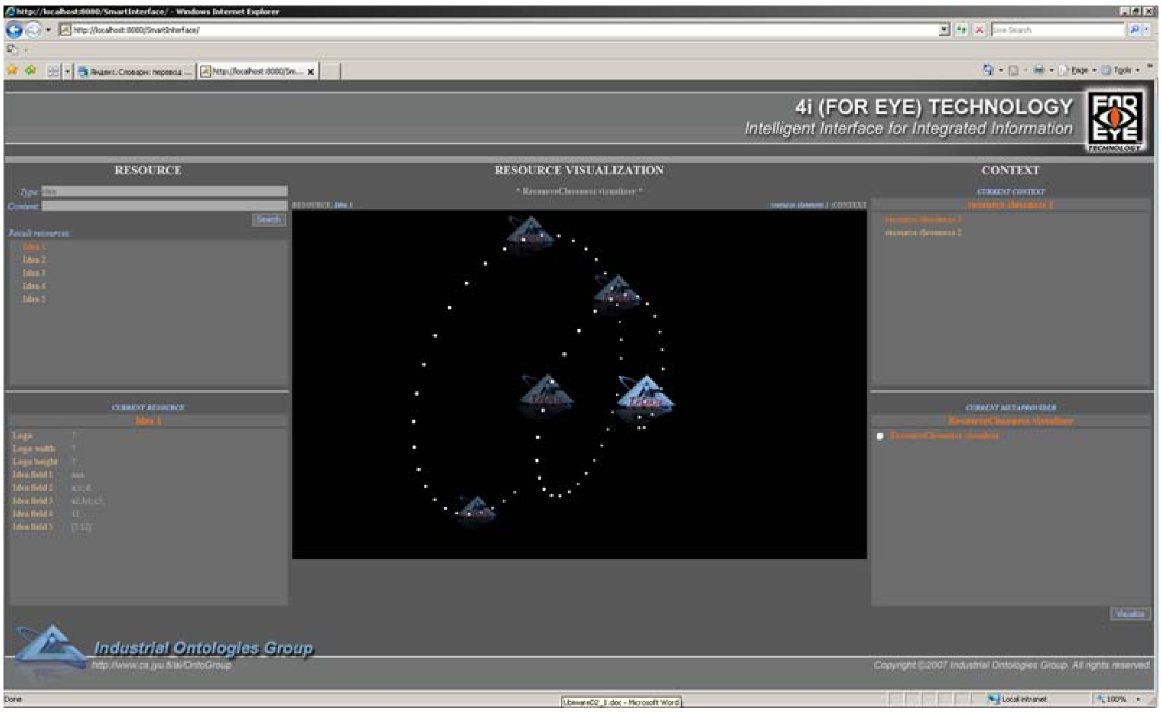


Figure 8 – resource closeness visualization in 4i (FOR EYE) Browser.

In this case we implemented and shown the flexible techniques for visualization context configuration. According to the general architecture of 4I (FOR EYE) Browser, adaptation of data sources is done via importing the source to the Browser and converting the data to the internal format. Current adapter works with both input formats: RDF with XML serialization and RDF in N-triple format. The data type of the source storage provided by Inno-W Company for the current industrial case is N-triple RDF.

ABB industrial prototype

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. Case architecture was pretty much the same as in Fingrid case for the first year.

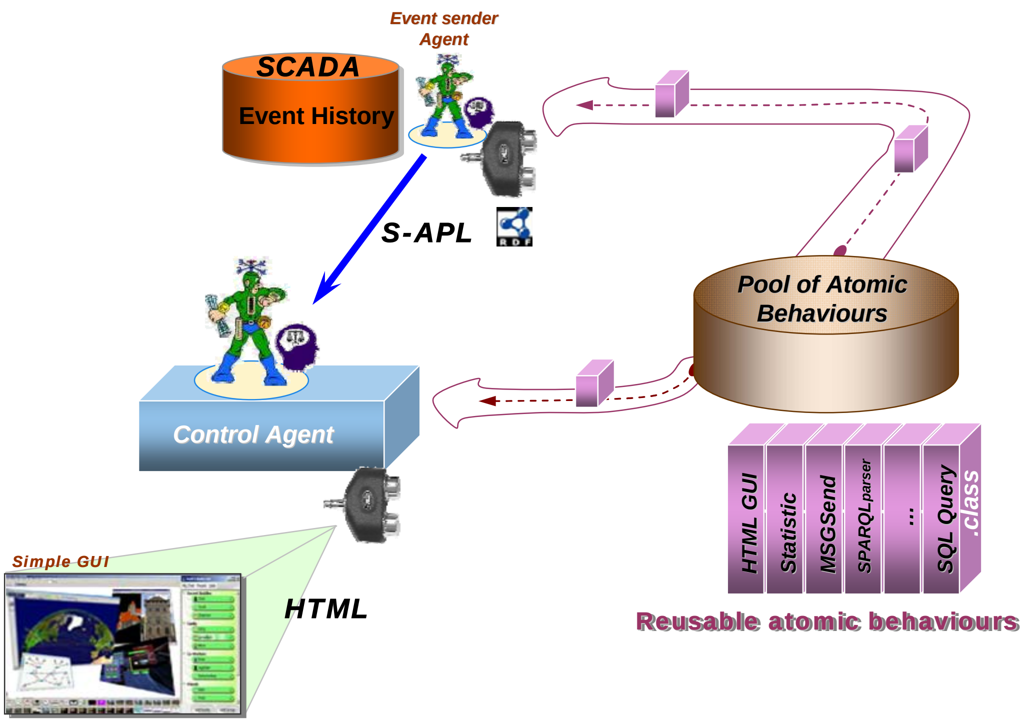


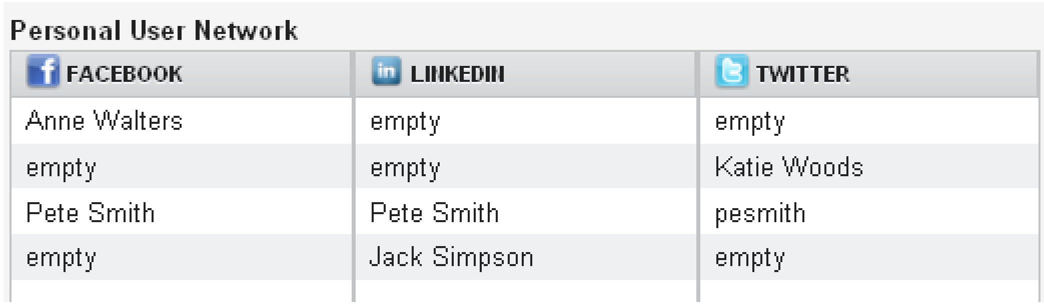
Figure 9 - Industrial prototype – ABB case.

IOG case: Mashupper – Agent-enabled Social Web

The Mashupper application uses data from three prominent social network platforms: Facebook, LinkedIn and Twitter. As the largest player in the field with more than 500 million users, Facebook is becoming a virtual world of its own. It fulfills the need for general socializing in the web and covers both leisure and work. LinkedIn on the other hand is very much profiled to the business and professional side of the social networking. Then there is the Twitter, which has capitalized on the people's need to hear and to be heard, preferably in real-time. The social connections in Twitter are looser than in LinkedIn or Facebook. Everything you write to Twitter (or tweet) is public and can be read by anyone. In Twitter, user can start to follow his or her friends or actually anyone who seems interesting enough, in order to receive updates from those people in real-time.

Facebook, LinkedIn and Twitter is a good set of services to start with, but there are many other interesting social networking sites out there. Adding new information sources to Mashupper is relatively simple task, thanks to the agent-based architecture of UBIWARE platform, given of course that the social network service provides some kind of API for external services.

The scope of the Mashupper is built on top of the concept of Personal User Network (PUN), which is defined as the combination of different kinds of human connections, which a particular user may have in the social web. *PUN is used to link the different online identities or profiles into one and same connection, through which the user can observe the integrated presence of that connection in the current web2.0 landscape*.



Personal User Network

FACEBOOK	LINKEDIN	TWITTER
Anne Walters	empty	empty
empty	empty	Katie Woods
Pete Smith	Pete Smith	pesmith
empty	Jack Simpson	empty

Figure 10 – Organizing PUN by linking profiles from different networks

In the current implementation, information sources are Facebook, Linked and Twitter. The retrieved information is shown using three different panels.

- o *Profile* panel shows the combined profile information about the selected person;
- o *Geo-location* panel uses Google Maps to display geo-tagged status updates on a map;
- o *Activity timeline* panel uses the Javascript widget from SIMILE (http://www.simile-widgets.org/timeline/) to visualize the stream status updates on a timeline.

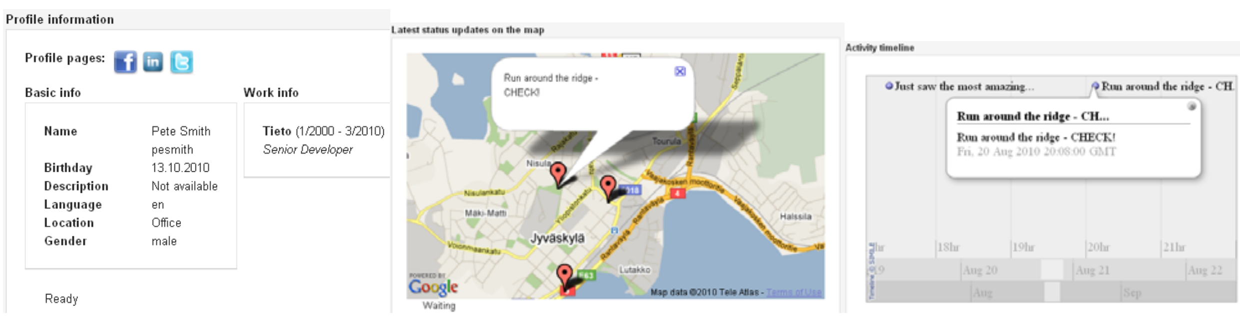


Figure 11 – Different panels from Mashupper interface.

Mashupper web application was developed using web development framework called Vaadin (http://vaadin.com/). Vaadin is a mature framework based on Google's web toolkit. It is developed and maintained by IT Mill company. Vaadin includes good collection of UI components for creating snappy looking web interfaces with relative ease. One of the main reasons why Vaadin was selected was the fact that the web applications can be written entirely in java, without having to bother with web related technologies such as HTML, CSS or Javascript. Vaadin made it easy especially for developers with experience in building Swing based Java applications.

By developing the Mashupper application we tried to demonstrate the applicability of the UBIWARE 3.0 platform to the variety of tasks, not necessarily related to the heavy industry domain. The easy implementation of the Social Networking demo has proven that UBIWARE is genuinely middleware solution – with globally broad scope of application areas. Whatever we develop within the platform, we can reuse further in absolutely different application cases and scenarios. The social networking blocks and components are now available as reusable parts for any new application case.

UBIWARE

Cooperation & Networking

The main international scientific collaborators of the group

- *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...

Scientific visits

2008

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.

PRIME-II project (FP7 / ICT)

2009

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.

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*

Cloud Software program (TIVIT SHOK) 2010

The main work of the Cloud Software program is done in three main work packages i.e. Key Business Driven initiatives. These work packages, Technologies in the Cloud, Lean Enterprise and Business in the Cloud, are industry led and most of the work of the Cloud Software program is done within these work packages.

The cloud is the future of software systems. Therefore, the main objective of WP1 is to address the technologies for developing software-intensive applications for the cloud, and on the cloud. This requires a deep understanding of the nature of applications as well as the new business models enabled by the cloud.

As a representative of the University of Jyväskylä in Cloud Software program (WP1), IOG perform the tasks that are aimed at integration of Soprano RDF framework with UBIWARE platform. In this project the main work is done for the Nokia and we the main cooperation is done with Nomovok (Subcontractor of Nokia). We are planning to continue this work also next year and show more benefits of UBIWARE platform for the tasks related to Cloud-based ecosystem development.

SCOPE project 2010

Another project that we are cooperating with is SCientific innOvation Product concEpt (SCOPE, 2009-2012) project run on the faculty of Information Technologies at the University of Jyväskylä. This project is supported by Nokia, EADS, NeedIt and other companies. Project is aimed to several aspects of mobile environment enhancements and new mobile application. Currently we are cooperation on one of them that concerns “Tactical platform to enhance public safety by situational awareness” (WP4). Within this cooperation we provide the information source integration and information matching solutions based on UBIWARE Platform.

cCloud project (Academy of Finland) 2010

IOG has been collaborating with finish academic partners from *University of Helsinki* and *Tampere University of Technology*. Together with these partners we submitted “cCloud: Collaborative clouds” project proposal to Academy of Finland as a result of this collaboration.

The cCloud project objective is to develop technologies and models for creation of open marketplaces and methodologies for the provision of service-ecosystem infrastructure elements. Such infrastructures provide service matching, eContracting, and ecosystem evolution services that can be utilized in coherent, SOA-based, and potentially agent-driven ways. Modules for automated, context- and content-sensitive interoperability solutions form part of this frame.

As a basement for such collaborative Ecosystem environment, we consider a network of the platforms that provide such main functionalities as cross-platform communication, interoperability of heterogeneous resources (components) and a toolbox for component composition on both levels. Due to heterogeneity of provided services and supported components, UBIWARE is based on integration of several technologies: Semantic Web, Distributed Artificial Intelligence and Agent Technologies, Ubiquitous Computing, SOA (Service-Oriented Architecture), Web X.0, and related concepts. Thus, we are going to use the platform as a basis and extend its functionality towards the needs of Cloud Ecosystem elaboration.

We consider this cooperation as a great opportunity for further utilization of the project results and dissemination of them on a Finnish academic level.

i-Cloud project (FP7/ICT)

2010

As a representative from University of Jyväskylä (Finland), together with IT'IS Foundation (Switzerland), TECHILA (Finland), SPEAG AG (Switzerland), University of Freiburg (Germany), EADS (Germany) IOG is going to participate in "i-Cloud" EU project, FP7/ICT – Cloud Computing.

SOFIA project proposal (TEKES)

2010

The project aimed to bring together new ICT approaches and technologies (Semantic Web, Autonomic Computing, and Internet of Things) which are amongst mainstream trends in information integration and engineering of complex systems to boost adoption of new business models in forest industry. The practical expected outcome of the project was a *SOFIA (Seamless Operation of Forest Industry Applications) software platform for B2B mediation tailored to the forestry sector of Finland*. We planned to have a strong national level network of forestry companies and organizations in order to get support and promote the new vision of flexible contracting for SMEs. Preliminarily we have discussed and received positive feedback from: organizations - Koneyrittäjät ry, Metsäalan kuljetusyrittäjät ry; forest industry companies and universities – Metsäteho, Metla, Joensuun yliopisto, VTT, Metsäkeskus Tapio, Metsäteollisuus ry, Energiateollisuus ry.

SERTE and D2I (TIVIT SHOK programs)

2010

Taking a part in two new TIVIT SHOK programs ("Services in Real Time Economy" and "From Data to Intelligence") is a perfect way to apply UBIWARE platform and new paradigm of middleware development in real industrial problem solving tasks.

Now, on a stage of program preparation, we are looking for the companies that are interested in our approach to create the consortiums and apply for the projects within the correspondent programs.

TECHILA

2010

We cooperate with Techila Company that provides cloud infrastructure solutions. The company has an agreement with a faculty of Information Technologies at the University of Jyväskylä and could provide a missing part (Infrastructure as a Service) for the Cloud Architecture based on UBIWARE Platform. We are preparing the project plan for Cloud Software program.

New specialists for industry

One more outcome of the project is five specialists that have been taught and have got correspondent skills from the UBIWARE project and now are working for industry in different local and international companies.

Internationalization

The project was national and has involved industrial and academic partners only from Finland. But, even then it has an international impact in a sense that project had a very international team. Researchers from 9 countries have been involved into research and development process during the project years.

IOG team has got an award as the most international team (Agora Center, University of Jyväskylä).

UBIWARE

The Main Publications

Edited Books and Book Chapters

1. Terziyan V., Katasonov A., Cardoso J., Hauswirth M., Majumdar A., PRIME: Proactive Inter-Middleware for Global Enterprise Resource Integration, In: E. Õunapuu (ed.), *Agile and Self-Organizing Enterprise Information Systems: Developing a Cloud Platform*, IGI Global, 2011, 22 pp. (*submitted*).
2. Kesäniemi J., Resources-Oriented Infrastructure for MAS Environment, *Multi-Agent Systems*, IN-TECH Publishing, Vienna, Austria, ISBN 978-953-7619-X-X (*submitted*).
3. Katasonov A., Terziyan V., Using Semantic Technology to Enable Behavioural Coordination of Heterogeneous Systems, In: Gang Wu (ed.), *Semantic Web*, IN-TECH Publishing, Vienna, Austria, January 2010, ISBN 978-953-7619-54-1, pp. 135-156. (Chapter VIII).
4. Nagy M., Katasonov A., Khriyenko O., Nikitin S., Szydlowski M., Terziyan V., Challenges of Middleware for the Internet of Things, In: A. D. Rodi (ed.), *Automation and Control - Theory and Practice*, IN-TECH Publishing, Vienna, Austria, December 2009, ISBN 978-953-307-039-1, pp. 247-270. (Chapter XIV).
5. Terziyan V., Katasonov A., Global Understanding Environment: Applying Semantic and Agent Technologies to Industrial Automation, In: M. Lytras and P. Ordonez De Pablos (eds.), *Emerging Topics and Technologies in Information Systems*, IGI Global, 2009, ISBN: 978-1-60566-222-0, pp. 55-87 (Chapter III).
6. 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.

Journal Papers

7. Terziyan V., Kaykova O., Zhovtobryukh D., UbiRoad: Semantic Middleware for Cooperative Traffic Systems and Services, In: *International Journal on Advances in Intelligent Systems*, Vol. 3, No 34, 2010, 14 pp. (to appear).
8. 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.

Refereed International Conference Papers

9. Khriyenko O., Nikitin S., Terziyan V., Context-Policy-Configuration: Paradigm of Intelligent Autonomous System Creation, In: Joaquim Filipe and Jose Cordeiro (Eds.), *Proceedings of the 12th International Conference on Enterprise Information Systems (ICEIS-2010)*, 8-12 June, 2010, Funchal, Madeira - Portugal, ISBN: 978-989-8425-05-8, pp. 198-205.
10. Nikitin S., Terziyan V., Nagy M., Mastering Intelligent Clouds: Engineering Intelligent Data Processing Services in the Cloud, In: *Proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics (ICINCO-2010)*, 15-18 June, 2010, Funchal, Madeira - Portugal, 8 pp.
11. Nikitin S., Terziyan V., Lappalainen M., SOFIA: Agent Scenario for Forest Industry, In: *Proceedings of the 12th International Conference on Enterprise Information Systems (ICEIS-2010)*, 8-12 June, 2010, Funchal, Madeira - Portugal, 8 pp.
12. Terziyan V., Kaykova O., Zhovtobryukh D., UbiRoad: Semantic Middleware for Context-Aware Smart Road Environments, In: *Proceedings of the Fifth International Conference on Internet and Web Applications and Services (ICIW-2010)*, May 9-15, 2010, Barcelona, Spain, IEEE CS Press, 8 pp.
13. 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, 6 pp.
14. 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.

15. 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, ISBN: 978-0-88986-800-7, pp. 184-191.
16. 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, 7 pp.
17. Terziyan V., Zhovtobryukh D., Katasonov A., Proactive Future Internet: Smart Semantic Middleware for Overlay Architecture, In: *Proceedings of the Fifth International Conference on Autonomic and Autonomous Systems (ICAS 2009)*, April 21-25, 2009, Valencia, Spain,IEEE,6 pp.
18. Katasonov A., Terziyan V., 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.
19. 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.
20. 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.
21. 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.
22. 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.
23. 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
24. 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)*, Barselona, Spain, 28-31 July 2007.
25. 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.
26. 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, pp. 382-389.
27. 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.

Academic Degree Theses

28. Nikitin S., Dynamic Aspects of Industrial Middleware Architectures, *PhD. Thesis, Department of Mathematical Information Technology,* University of Jyväskylä, January 2011.
29. Pilli-Sihvola V., Intelligence as a Service, *MSc. Thesis, Department of Mathematical Information Technology*, University of Jyväskylä, December 2010.
30. Bleier A., A Framework for Market-Based Coordination in Multi-Agent Systems, *MSc. Thesis, University of Osnabruck and Department of Mathematical Information Technology*, University of Jyväskylä, September 2008.
31. Khriyenko O., Adaptive Semantic Web based Environment for Web Resources, *PhD. Thesis, Department of Mathematical Information Technology,* University of Jyväskylä, Vol. 97, Dec. 2008.
32. Naumenko A., Semantics-Based Access Control in Business Networks, *PhD. Thesis, Department of Mathematical Information Technology*, University of Jyväskylä, Vol. 78, June 2007.

UBIWARE

Utilization of Project Results

Use in industry and other projects

Currently, UBIWARE Platform is used by Fingrid Company. The industrial case with Fingrid has been running as part of Fingrid operative environment for some time and is being exploited by over 20 specialists within the company. The application is responsible for data integration from several heterogeneous sources, data presentation, anomaly detection and anomaly reporting.
Also, UBIWARE Platform is used for solving the information source integration and information matching purposes within a workpackage WP4:"Tactical platform to enhance public safety by situational awareness" in cooperation with SCOPE (SCientific innOvation Product concEpt) project.

Courses

Based on achieved project results and developed UBIWARE Platform, IOG team have prepared and taught three courses:

- *"Design of Agent-based Systems"* - The course focuses on the use of Distributed Artificial Intelligence methods, and more specifically of Intelligent Agents Technologies, for development of complex distributed software systems.
- *"Semantic Web and Ontology Engineering"* - The course focuses on emerging Semantic Web and intelligent information integration technologies applied for commercial applications in mobile environment. Course includes an introduction to Semantic Web knowledge markup techniques and markup languages RDF-based semantic annotation of Web resources and services, ontology engineering. Course also considers modern applications of these methods and techniques for Web-based intelligent applications and services.
- *"Service-oriented Architectures and Cloud Computing for Developers"* - The course focuses on proper techniques for web service development and composition by following SOA (Service oriented architectures) approach. Topics related to semantic SOA are touched as well. In addition to that, the course deals with cloud computing paradigm, mostly from software engineering point of view. The course is planned for the next semester and will be started in the beginning of the year 2011.

UBIWARE Platform has been tested with students as a tool for creating distributed autonomous systems as part of the course work. It helps students to better understand and apply new theoretical achievements of the courses.

Future plans of use

Together with academic partners from *University of Helsinki* and *Tampere University of Technology* we submitted "cCloud: Collaborative clouds" project proposal to Academy of Finland. Within this project we are planning to utilize UBIWARE platform as a basis and extend its functionality towards the needs of Cloud Ecosystem elaboration. We consider this cooperation as a great opportunity for further utilization of the project results and dissemination of them on a Finnish academic level.

As a future continuation of work in Cloud Software program (TIVIT SHOK), together with TECHILA Company we are preparing the project plan for a new Cloud Architecture based on UBIWARE Platform.

Now we are also trying to be involved to new TIVIT SHOK programs: "Services in Real Time Economy" (SERTE) and "From Data to Intelligence" (D2I). We consider this as a perfect way to apply UBIWARE platform and all other achievements of the project for solving real industrial problems. Now we are looking for the companies that are interested in our approach to create the consortiums and apply for the projects within these programs.

On international level, together with consortium of partners from Switzerland, Germany and Finland, we are going to utilize platform in "i-Cloud" EU project, submitted to FP7/ICT – Cloud computing program.

UBIWARE

Project Budget

	2007	2008	2009	2010	Total Expenses*	Planed Budget
Salary & Overheads:	160 258	390 992	248 662	335 030	1 134 942	1 135 532
Travels:	9 163	10 292	7 958	9 033	36 446	37 833
Equipment & Consumables:	4 162	1 345	75	53	5 635	5 635
TOTAL (Euro):	173 583	402 629	256 695	344 116	1 177 023	1 179 000

Funding from companies & TEKES:

	2007	2008	2009	2010	Total Expenses*	Planed Budget
Metso Automation:	20 000	20 000	20 000		60 000	60 000
Fingrid:	10 000	20 000	20 000	10 000	60 000	60 000
Inno-W:	500	1 000	12 000	2 500	16 000	16 000
Hansa Ecuras:	own work	own work	own work	own work	own work	own work
ABB:	10 000	10 000			20 000	20 000
Nokia:			20 000		20 000	20 000
Metso Shared Services:	3 000				3 000	3 000
TEKES:	138 693	321 700	217 702	319 928	998 023	1 000 000
TOTAL (Euro):	182 193	372 700	289 702	332 428	1 177 023	1 179 000

UBIWARE

Summary

Summary

The UBIWARE project is a major step in a longer path that aims to build the so called global understanding environment. That is, a platform or middleware that supports flexible integration of all kinds of resources that have not been a priori designed to be interoperable into new processes that have not been specified when designing the platform. The main project outcomes can be classified into three groups that are aimed to different audiences: concrete industrial pilots and applications, a platform for application design and a new paradigm in software engineering. Concrete progress has been made in all three main directions.

In this project we introduced new paradigm in software engineering and elaborated approach towards creation of semantically enhanced agent-based integration middleware that makes heterogeneous resources proactive, goal-driven and able to interoperate with each other in collaborative environment.

One of the main theoretical results of the UBIWARE is *Semantic Agent Programming Language (S-APL)*, which is a Resource Description Framework (RDF) - based language integrating features of several kinds of tools: agent programming languages (like AgentSpeak and AFAPL), semantic reasoners (like CWM), querying languages (like SPARQL) and agent communication content languages (like FIPA SL). The S-APL engine was implemented and is central component of UBIWARE Platform and provides a base that is being used now for realizing all other UBIWARE functionality.

We have introduced Ontonut technology to tackle the problem of distributed querying in UBIWARE-based multi-agent systems. Due to *ontonuts* technology and based on the *Open World Assumption* (alternatively to BPEL with the Closed World Assumption), S-APL engine is able to discover and automatically utilize (in a runtime) new services (capabilities) that have just appeared in semantic service registry. We also 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. We described our long-term vision for the security and privacy management in such complex environments, SURPAS (Smart Ubiquitous Resource Privacy and Security), and have treated questions, related to policies, privacy and security on the more general level of organizational policies of any kind.

The *Service Oriented Architecture* (SOA) is an approach to integrating available enterprise applications in a flexible and loosely coupled manner to enable more sophisticated, complex and distributed applications. SOA is built on the notion of services (external capabilities, which are realizations of self-contained business functions). *SOA* is based on *choreography* and *orchestration* of services. *Choreography* is concerned with describing the external visible behavior of services, as a set of message exchanges, from the functionality consumer point of view. *Orchestration* deals with describing how a number of services, two or more, cooperate and communicate with the aim of achieving a common goal. *S-APL* is a language capable to describe both choreography and orchestration (through *ontonuts*) of external capabilities (data or functional services) and internal atomic capabilities (*Reusable Atomic Behaviors*) needed for designing and executing a complex business process.

With a respect to human-centric technologies, we also proposed an approach of semantically enhanced context-dependent multidimensional resource visualization that provides an opportunity to create intelligent visual interface that presents relevant information in more suitable and personalized for user form. Proposed 4I (FOR EYE) technology (Intelligent Interface for Integrated Information) can be considered as a next step towards new generation of context-aware service-oriented collaborative user interfaces.

The basic approach in development has been an agile development – creation of a succession of prototypes with improving functionalities on every release combined with concrete use cases with companies. The core of the UBIWARE platform gives 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 enables the resources to automatically discover each other and to configure a system with complex functionality based on the atomic functionalities of the resources. It ensures 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.

We have brought the UBIWARE platform to the qualitatively new level of the middleware solution – the platform now combines the features of the application server, the semantic web platform and the agent-driven platform, where agent-driven semantic applications can serve end customers with the high quality web-based GUIs, enhanced user-friendliness and responsiveness. The platform has become an application-independent runtime environment, where special infrastructure agents take care of the platform itself, not of the applications being run on it. At the same time, we introduce personal user agents, thus making the platform user-oriented. We design the UBIWARE platform infrastructure for creation of various kinds of applications. Those applications have a freedom to use a web front-end, on-the-platform user management and other infrastructure or define their own platform components depending on the needs of the application.

The new paradigm in software engineering that has emerged during the development has been found working and useful both in platform related development as well as in software development that is not dependent on UBIWARE platform. The paradigm (of exploiting fully the semantics related to the use cases and resources) has been used in all company cases. During the years of the project we have been developing five industrial cases for Fingrid, Metso Automation, Inno-W, ABB and Nokia Companies. The Inno-W case is intended to be an independent application outside the portal but its design relies heavily on using the semantic descriptions related to the content, operational context and external resources that are used to build the application. We developed a prototype of 4I Browser to proof the concepts behind proposed 4I (FOR EYE) technology. 4I Browser is a kind of mashuper of context-dependent visualizers, it is a smart ensemble of Intelligent GUI-Shell (smart middleware for context dependent use and combination of a variety of different MetaProviders, depending on the user needs) and MetaProviders as graphical interfaces that visualize filtered integrated information. In portal related development the concept of ontonut (arising from the Metso case) relies on the same approach, exploiting the rich semantic description of the resources (data, services) on one hand and goals and capabilities of the agents living on the portal on the other hand. At the same time, there are several developed applications that do not belong to the companies' needs (*Mashupper* – Agent-enabled Social Web). They are developed as a proof of concepts behind a certain functionality of the Platform.

The UBIWARE platform itself has been now used to run several test cases, extended in various ways with new capabilities based on the research findings made in the project. Moreover, it has been tested with students as a tool for creating distributed autonomous systems as part of the course work. However, the platform has not been augmented so far with user interface tools etc. that would better support its use in software engineering. Hence the platform is still a prototype, not really a software product that could be exploited easily outside the project group. Team has a clear vision on further improvement and enhancement of the platform. UBIWARE 4.0 will be finished after the project. It will include scenario editor, better user-as-developer interface, and payment system as an internal reusable functionality and will be open for updates. Current version of the platform is presented as a working cloud in the web and all the related to the project materials and reports are available on a correspondent project web-page through the website of IOG[3].

Currently, one industrial case (based on UBIWARE Platform) is actively used by Fingrid Company. The case has been running as part of Fingrid operative environment for some time and is being exploited by an increasing number of users within the company. Theoretical project achievements and platform are used in Cloud Software project (TIVIT) and in cooperation with SCOPE project as well. We are going to use them in “cCloud: Collaborative clouds” project (proposed to Academy of Finland), submitted international “i-Cloud” EU project (FP7/ICT), and in two new TIVIT SHOK programs: “Services in Real Time Economy” (SERTE) and “From Data to Intelligence” (D2I).

[3] UBIWARE project web-page - http://www.cs.jyu.fi/ai/OntoGroup/UBIWARE_details.htm

Environment of the project was very assistive for the students that took part in the project research and development. As a result of it, project gave as outcome 3 PhD. and 2 M.Sc. academic degrees. Based on achieved project results, UBIWARE team have prepared and taught three courses: "*Design of Agent-based Systems*", "*Semantic Web and Ontology Engineering*" and "*Service-oriented Architectures and Cloud Computing for Developers*" (is planned for the next semester). About 30 scientific publications (book chapters, journal and conference papers) have been published during the project years. One more outcome of the project is five specialists that have been taught and have got correspondent skills from the UBIWARE project and now are working for industry in different local and international companies.

References

References

1. 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.
2. 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.
3. 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.
4. Khriyenko O., and Terziyan V. (2006), A Framework for Context-Sensitive Metadata Description, In: International Journal of Metadata, Semantics and Ontologies.
5. 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.
6. 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.