UbiRoad: Semantic Middleware for Cooperative Traffic Systems and Services

Vagan Terziyan
MIT Department, University of Jyvaskyla
P.O. Box 35 (Agora), 40014
Jyvaskyla, Finland
e-mail: vagan@jyu.fi

Olena Kaykova, Dmytro Zhovtobryukh
Agora Center, University of Jyvaskyla
P.O. Box 35 (Agora), 40014
Jyvaskyla, Finland
e-mail: olena@cc.jyu.fi, dzhovto@cc.jyu.fi

Abstract—**Emerging traffic management systems and smart road environments are currently equipped with all necessary facilities to enable seamless mobile service provisioning to the users. However, advanced sensors and network architectures deployed within the traffic environment are insufficient to make mobile service provisioning autonomous and proactive, thus minimizing drivers' distraction during their presence in the environment. An ideal system should provide solutions to the following two interoperability problems: interoperability between the in-car and roadside devices produced and programmed by different vendors and/or providers, and the need for seamless and flexible collaboration (including discovery, coordination, conflict resolution and negotiation) amongst the smart road devices and services. To tackle these problems, in this paper we propose UbiRoad middleware intending utilization of semantic languages and semantic technologies for declarative specification of devices' and services' behavior, application of software agents as engines executing those specifications, and establishment of common ontologies to facilitate and govern seamless interoperation of devices, services, remote systems and humans.**

Keywords- context-aware services; cooperative traffic; smart road; middleware; semantic technologies; agents

I. INTRODUCTION

There is about half of a billion drivers only in Europe, who wish driving to be more comfortable, efficient, ecological and less risky. Not far are the times when cars will themselves prevent accidents. People spend more time in vehicles and they are expecting also more possibilities to work and use various services while traveling, which requires new travel infrastructure and automation services [2]. These should combine various vehicles, their drivers and passengers, smart roads and appropriate Web services [3]. Recent wireless and internet technologies enable completely new possibilities to integrate available efforts into the new advanced traffic paradigm – cooperative traffic [4].

Service-oriented architectures related to traffic management, smart roads and future context-aware services for drivers are closely integrated into the Internet of Things [5], which is a world where things can automatically communicate to computers and each other, providing services for human benefits. In such "Future Internet", intelligence and knowledge will be distributed among an extremely large number of heterogeneous entities: sensors, actuators, devices, cars, road infrastructures, software applications, Web services, humans, and others. To realize this vision, there is a need for an open architecture, which will offer seamless connectivity and interworking between these heterogeneous entities. Moreover, ensuring collaboration, synchronization but also control of this distributed intelligence is a challenge that needs to be addressed, or the Internet of Things will become a chaotic, un-controlled and possibly dangerous environment since some actors of this Internet have impact on the real world (e.g., software or humans through actuators). Cooperative traffic domain enables interoperability between a large number of heterogeneous entities, while ensuring predictability and safety of their operation, is difficult without an extra layer of intelligence that will ensure the orchestration of these various actors according to well-defined goals, taking into account changing constraints, business objectives or regulations. This paper introduces such a middleware layer (UbiRoad). It provides cross-layer communication services (data-level interoperability) to the entities and extended multi-agent technologies will provide collaboration-support services (functional protocol-level interoperability and coordination) for these entities. The UbiRoad middleware concept apparently entails a vision of a multifaceted, multi-purpose and multipronged middleware platform applying multidisciplinary approach to extension and enhancement of the future smart traffic environments UbiRoad middleware should be rather seen as a meta-structure on top of the future intelligent transportation systems and services and as intelligent stratum between the smart road device layer and the future service oriented architectures.

A first major problem to be addressed by UbiRoad is inherent *heterogeneity*, with respect to the nature of components, standards, data formats, protocols, etc., which creates significant obstacles for interoperability among the components of ubiquitous computing systems. This heterogeneity is likely to induce some integration costs that will become prohibitive at a very large scale preventing a rich ecosystem of applications to emerge. It is generally recognized that achieving the interoperability by imposing some rigid standards and making everyone comply could not be a case in open ubiquitous environments. Therefore, the interoperability requires existence of some middleware to act as the glue joining heterogeneous components together.

The second major issue is to guarantee high level of *safety*. Since the IT infrastructure and through them users are going to have real actions in the real physical world through various actuators we have to ensure that these actions are properly controlled and coordinated. Despite the wish to enable as many actors as possible to have access to physical world objects around the world to enable a large set of diverse applications, this should be done in a well-understood and safe manner. The “things” will have to exhibit some required behaviors that humans have adopted to assemble in cooperative traffic social interactions.

The UbiRoad approach can be seen as studying the triangle of device-software-human interaction seen from the perspective of the above described scenarios. Henceforth we refer to “device” as to any monitored or controlled physical objects including e.g., vehicles. Substantial research results related to edges and vertices of this triangle have been (recently) reported [6, 7, 8] (e.g., efforts related to middleware for embedded systems, efforts related to integration of diverse software systems and services, etc). What is missing is an integrated coherent approach to cover the whole triangle. Moreover, many on the past research initiatives do not truly deal with the core topic, which is *interoperability versus just interconnectivity*. The components of cooperative traffic systems should be able not only to communicate and exchange data, but also to flexibly coordinate with each other, discover and use each other, learn about the location, status and capabilities of each other, and jointly engage in different traffic situations. Moreover, the components must achieve the above using an always-on, safe, robust and scalable means of interaction.

Further in this paper, we argue in favor of fully interoperable (though heterogeneous), highly dynamic and extensible smart road environments. We present a specialized agent-driven middleware platform UbiRoad, in which each ubiquitous smart device (as well as each individual service exposed as an individually accessible entity through the environment) will be assigned a representative agent within UbiRoad. The resulting multi-agent system will be exploited as a mediation facility enabling rich cooperation capabilities (e.g., discovery, coordination, adaptability, and negotiation) amongst the devices inhabiting the smart traffic environment. Utilization of semantic technologies [9] in UbiRoad will ensure efficient and autonomous coordination among UbiRoad agents and will thus ensure interoperability between associated devices and services. Several UbiRoad ontologies are an important asset contributing to interoperability realization within future smart traffic environments. These ontologies are used not only for the benefit of UbiRoad middleware architecture, but also and most importantly for facilitation of interoperability and integration of existing and brand-new future devices, services and methodologies. Through appropriate declarative specification of smart road components’ behavior and using sophisticated choreographic control agents in a multimodal dynamic networked environment, the UbiRoad enables various devices and services to automatically discover each other and to configure complex services functionally composed of the individual services’ and devices’ functionalities.

The rest of the paper is organized as follows: in Chapter II we are providing the motivating scenario for the new challenging requirements to traffic management systems; in Chapter III we list the requirements and related challenges to be addressed when designing such systems; in Chapter IV we provide possible solution for the challenges based on the concept and architecture of the so called Global Understanding Environment; in Chapter V we discuss some important and challenging features of appropriate agent-driven platform (UBIWARE) suitable for UbiRoad implementation, such as: semantic adapters and integrators (called *OntoNuts*); semantic visualization technology (called *for-eye*); user-driven system configuration (via so called *smart comments*); and *semantic blogging*; in Chapter VI we overview appropriate software architecture; in Chapter VII we briefly discuss on Traffic and Mobility ontology and system integration; in Chapter VIII we briefly comment on related work; and we conclude in Chapter IX.

II. MOTIVATING SCENARIO

Consider the following story, in which we try to integrate several possible scenarios of future use for the UbiRoad middleware.

(*Beginning of the story*) “Timo lives in Jyväskylä. Former researcher, he is a widely recognized expert in the field of intelligent software agents. Nowadays Timo owns a small IT business based in Jyväskylä, and his firm is often subcontracted by large IT and telecom enterprises to perform highly specialized development services. Therefore, Timo is a frequent guest in Helsinki and Helsinki region, where most of his company’s employers reside. Despite considerable distance between Jyväskylä and Helsinki, Timo likes neither airplanes, nor trains, and always travels inside Finland by car. Fortunately, he is a big cars-lover and a good driver.

Timo arrived in Helsinki early in the morning and spent the whole day participating in a few various business meetings and research seminars. Now, when he is about to leave Helsinki, he feels very tired. He could stay in Helsinki overnight, but he has another important meeting scheduled for tomorrow at 7 am in Jyväskylä. Not to fall asleep on the way back to Jyväskylä, Timo drops by the nearest cafeteria and drinks a cup of strong coffee. Having felt a burst of energy after the sprightly drink, he gets into his car and leaves Helsinki at dusk. In the car Timo selects from his audio collection some nice invigorative music to listen to and sets the car control system’s operating mode to ‘exhaustion’ using the available on-board control panel. Timo knows that in this mode the awareness levels of a multitude of software agents, which inhabit his car and make it a part of the UbiRoad intelligent transportation system, reach the highest possible value. Now he feels much less vulnerable because of fatigue, as agents in this mode help significantly reduce exposure to various on-road risks. The in-car control system adjusts climate conditions (temperature, humidity, level of oxygen) to optimal levels with respect to the selected operating mode: it aims to maintain cool, fresh, oxygen-rich atmosphere inside the car in order to prevent the driver from falling asleep; and activates on-board alarm system, which is configured to give to the driver light and audio indication every 30 seconds (not allowing him falling asleep). The corresponding UbiRoad traffic agent (representing Timo and his car as a dynamic road user entity) sets own hazard level to ‘red’, thus notifying other road users of potential risks associated with its road user. When Timo drives onto the motorway going out of Helsinki, he feels much more comfortable and relaxed, as he is sure that all necessary measures of passive risk prevention have been undertaken and as he should no longer pay attention to the oncoming traffic (on motorways directions of traffic are separated).

To shake himself up a bit, Timo switches to the left lane (each direction of the motorway connecting Helsinki and Lahti has two lanes) and starts overtaking all cars, which slowly go on the right lane. The speed limit on this motorway is 120 kmph and driving at it can be refreshing. After some time of such racing Timo however forgets about the speed, which immediately goes beyond 130 kmph, and the traffic agent monitoring velocity and controlling speed regimes detects inadmissibly excessive speed (via comparing actually measured vehicle’s velocity with the speed limit that the agent can read from RFID-enhanced traffic signs located on the sides of the road) and activates a loud beep tone combined with an appropriately marked blinking red LED on the control panel. Timo takes this as a timely signal to calm down, decelerates to the allowed speed limit and uses the cruise control functionality embedded into the steering wheel to fix the speed at the current level. Now he can release the accelerator completely and give his leg some rest.

Timo utilizes voice control system to engage a travel estimator service and thus to find out the approximate time of his arrival in Jyväskylä. The specialized voice recognition system reads Timo’s oral instructions, interprets them and finally transforms them into the format recognizable by UbiRoad agents. Then the corresponding communication agent finds an appropriate travel estimator service in the Internet, negotiates service contract with the agent representing the service and finally invokes the service. The result of travel duration estimation, 2 hours and 10 minutes, appears on the LCD screen built in the control panel to the right of the driver’s seat. Timo decides to call home and let his wife know he is coming back soon. Timo’s mobile phone is already connected with the in-car control system via Bluetooth. Timo utilizes voice control to access his phone and then voice dial to call Anna. While the picked number is being dialed, playing music is automatically damped down, and as soon as the phone connection is established, the conversation is output through the in-car embedded speaker system. After several minutes of chatting with Anna, Timo notices that he is driving already in the neighborhood of Lahti. Here 3G telecommunication network is available. The communication agent immediately detects this and using the LCD screen asks Timo if he is willing to switch to a video call. Timo accepts the offer by pressing the corresponding button on the touch-sensitive screen. The communication agent immediately requests the video capture service from a tiny camera embedded in the control panel in front of the driver’s seat. Then it rearranges the current voice call session as a new video call session without interrupting the call and interweaves the audio component acquired through Timo’s hands-free microphone with the video component obtained by the in-car embedded video camera. A live view of Anna appears on the LCD screen of the control panel. However, as shifting driver’s focus to this side screen is inconvenient and distracting the driver from actual driving, the picture on the screen is instantly projected on the internal surface of the car’s windscreen just in front of the driver’s seat. The projected image is however semi-transparent not to impede driver’s clear view of the road.

Timo finishes talking with his wife when Lahti is already left behind. He notices that twilight almost gave the place to solid night, but the motorway is still well illuminated. Timo decides to make a short stop at the picturesque roadside restaurant “Tähtihovi” in order to stretch his legs and have another cup of coffee before proceeding to the most difficult and boring part of his trip. Soon after this stop Timo should drive off the motorway to the side route leading to Jyväskylä. The traffic agent recognizes this major route change and reminds Timo of it well in advance using available visual indication means (LCD screen, projection on the windscreen, etc.) As Timo turns to the needed side road, he soon finds himself completely benighted as roadside lamps are uncommon here. He switches to upper beam to see at least something. Using embedded luminosity sensors, the agent monitoring external physical environment immediately detects severe lack of light on the road and activates built-in night vision system that multiply amplifies luminosity of the reflected light both in visible and infrared spectrum, thus being able to identify distant objects also by the heat they emit (e.g., oncoming cars, cyclists, pedestrians, elks, etc.). Such enhanced view of the road environment is projected on the internal surface of the car’s windscreen so that it maximally coincides with the driver’s field of view. Hence, Timo is now able to see everything much more clearly and recognize moving objects well in advance. What is more, in observed conditions of dark driving on a narrow bidirectional road the traffic agent starts to provide necessary assistance services such as improved navigation and automated signaling, e.g., a dynamically changing light-modulated traffic map of the neighborhood (specifically highlighting the route undertaken) is projected on the right side of the windscreen; upcoming turns and bends of the road are visually indicated (e.g., in the form of light arrows in the upper part of the windscreen); crossroads and cars approaching from the opposite direction are also identified for the driver in good time; switching from upper to lower beam (in proximity of oncoming cars) and back is performed automatically.

Luckily, the road is almost empty at night, and Timo almost reaches Jyväskylä when he catches up a heavy truck slowly going ahead of his car. Road is constantly dodging and the road-bed is narrow to comfortably overtake the truck. Timo almost loses patience waiting for a more or less straight section of the road, and as soon as such section appears ahead, he confidently sends the car on the opposite lane and starts overtaking the truck. Suddenly he sees an opportune notification of an oncoming vehicle, which is still on the other side of the hill ahead of Timo and is thus unseen, but is quickly approaching. Perhaps, Timo is too tired as he makes an estimation error: he decides that he has enough space and time to complete the maneuver and continues overtaking. The oncoming car is however approaching too fast making head-on meeting with Timo's car almost inevitable. Moreover, the truck being overtaken turns out to be a long road-train, and it is already too late to get back behind it because Timo's car has passed more than a half of the truck's length already, when Timo realizes that he fell a victim to his own fatigue and impatience, and that only a miracle can now save him from head-on collision with the other car. UbiRoad intelligence is such a miracle.

The UbiRoad traffic agent that resides in Timo's car establishes communication with the approaching car's traffic agent immediately after it recognizes the presence of another vehicle in the proximity. At the same time it maintains communication with the traffic agent of the truck. The agents jointly monitor the process of rapprochement of the (three) vehicles. When Timo starts his overtaking maneuver, the traffic agents realize the situation is no longer standard. They integrate their individual traffic information, jointly reason upon it in the dynamic traffic context, and deduce that the collision is unavoidable. To prevent the traffic accident or any other dire consequences of Timo's mistake, the agents must undertake active measures of risk mitigation. The traffic agents of the approaching car and the truck notify their drivers of the potentially critical hazardous traffic situation and forcibly decelerate their vehicles to buy Timo enough time for successful completion of the overtaking maneuver. For its part, Timo's traffic agent aggressively visualizes the imperative "complete the maneuver", thus granting some extra confidence to its driver, who is already close to panic. Given such clear instruction, Timo accelerates even more and safely completes the overtaking maneuver. In twenty minutes, when he, exhausted as a squeezed lemon, but happy to escape probably fatal traffic accident, parks his car in his parking slot, another in-car agent reads Timo's schedule for tomorrow (stored in the organizer application within Timo's mobile phone) and sets engine warming-up timer to 6.30 am …" (*end of story*).

To be able to make this scenario a reality we have to face several challenges described in the next chapter.

III. UBIROAD MIDDLEWARE CHALLENGES

A. Interoperability

By proclaiming interoperability as its major ultimate objective, UbiRoad approach deals with three major types of interoperability problem: technical interoperability (being the capability of devices, protocols and other technical standards to co-exist and interoperate), semantic interoperability (being the capability of various system components to treat and interpret exchanged data and information identically and share a common understanding of it), and pragmatic interoperability (being the capability of system components to capture willingness of partners to collaborate or, more generally, to capture their (and even human users') intent). Technical interoperability will be achieved through the agent-based mediation between different devices and standards with the aid of special adapter components and tunneling mechanisms. Semantic interoperability is the main focus of the UbiRoad approach as it is a prerequisite for seamless information internetworking and integration, and for smooth autonomous communication between various resources within a smart traffic environment. Semantic interoperability can be achieved by exploitation of rich metadata describing informational objects and semantic resource descriptions written in compliance with well-established semantic standards and on the base of predefined domain ontologies and UbiRoad Ontologies. Pragmatic interoperability amongst smart space components is achieved through appropriate design of declarative specifications of such components' behavior and on-the-fly agent-based identification of this behavior using given descriptions. Finally, the most innovative type of interoperability, which UbiRoad provides, is the so-called 'cross-layer' interoperability, e.g., interoperability between devices and services in a smart traffic environment. This particular class of interoperability problems is often difficult to solve even on individual basis. However, UbiRoad provides native support for cross-layer interoperation by implementing the paradigm of resource-oriented networking. This paradigm enforces unified treatment of various system components, e.g., devices, services, applications and even users, as different types of resources (Figure 1).

Figure 1. Agent-driven smart road interoperability

The communication is then established between resources regardless their particular type provided that negotiation is performed by resources' representing agents (associated with resources within smart traffic environments and beyond) as shown in Figure 1 and appropriate Semantic Web standards for unified resource description are used.

B. Flexible Coordination

As smart traffic environments are basically deployed to provide users with dynamically configured, customized, value-added and on-the-move autonomously operating services, UbiRoad targets establishment of such service creation and provisioning framework that would emphasize the above mentioned characteristics of ubiquitous services. Customization, personalization, added value, dynamicity and autonomy of services is to be achieved through construction and utilization of context-aware, adaptable and reconfigurable composite service networks. Service networks can be composed using declarative specifications of service models. Reconfigurability of service networks is made possible via utilization of hierarchical modeling of service control and its run-time execution. Dynamic adaptation of services is performed by special context-aware control components built in service networks. The traditional tradeoff "customization vs. autonomy" can be dealt with through a balanced use of user-aware goal-driven on-demand service composition, AI-enriched active context-awareness capturing user intent, and user-collaborative passive context-aware service composition. Though it is a challenging task, utilization of agent-based approach for service composition makes it much more flexible compared to traditional orchestration approaches. This difference in flexibility can be seen from the definition of the traditional Semantic Web services (SWS) given in [18] ("Self-contained, self-described, semantically marked-up software resources that can be published, discovered, composed and executed across the Web in a task-driven way") and the definition of proactive (agent-driven) SWS given in [19] ("Self-contained, self-described, semantically marked-up *proactive* software resources that can be published, discovered, composed and executed across the Web in a task-driven way, *and which behave to increase their utility and are the subject of negotiation and trade*"). Agents can bring many valuable features into a service composition framework, e.g., precomposition, distributed hierarchical control of service networks (not requiring a dedicated underlying infrastructure), and enhanced negotiation of non-functional service parameters.

C. Self-Management

UbiRoad brings self-management aboard via presenting totally distributed agent-driven proactive management system. UbiRoad agents monitor various components, resources and properties within the system architecture and infrastructures belonging or otherwise interacting with the managed smart road environment, and react to changes occurred by reconfiguring the architecture in appropriate way with respect to the predefined (or inferred) configuration plan. Configuration plans basically represent enhanced business models, which are adhered to during accomplishment of communication procedures between different parties. Due to purely distributed layout of the agent system and outstanding agents' programmability, merely all kinds of business models can be formalized and enacted by the UbiRoad management platform (due to richness of the utilized agent communication language and of the associated ontology base). In addition to this, UbiRoad agents are capable of learning via utilizing available data mining algorithms and further dynamically reconfiguring the managed architecture on the basis of acquired knowledge, thus being capable of inferring (also collaboratively) new configuration plans. UbiRoad can be deployed on top of any architectural model (including ad-hoc and peer-to-peer, which is of crucial importance for highly dynamic traffic environments) due to benefits of agent technologies and open resource interfaces. Also, the UbiRoad platform can make use of contextual information extracted from the managed networking environment in order to act as appropriately to the observed requirements and circumstances as possible.

D. Trust and Reputation

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

E. Other Challenges

Specifically, due to utilization of extended intelligent agent technology UbiRoad significantly contributes to realization or enhancement of the following important characteristics and functionalities of collaborative traffic environments:

- Data mining and knowledge discovery (e.g., utilization of accumulated statistics of traffic accidents), which may be organized either by establishing centralized Web server with appropriate data processing services or by local processing of the analytics and exchanging of it in a P2P manner;
- Learning (e.g., case-based learning, when traffic agents can learn on sets of predefined examples of traffic situations);

- Global data and knowledge reuse (e.g., traffic environments have a common infrastructure, which inter alia provides means for storing and sharing of traffic information; agents may access external information sources located, for example, in the Internet);
- Enhanced traffic services (e.g., traffic services such as, for instance, traffic signs are RFID-annotated, which allows agents to identify them and conveniently communicate their meaning to drivers or take on appropriate actions);
- Enhanced collaboration between various road-users (e.g., collaboration between drivers on the road can be significantly enhanced and automated by the dialog between their representative agents; proximity-driven collaboration);
- Global context awareness, contextual filtering and visualization (e.g., traffic situations are treated by traffic agents in context (set of relevant contexts); traffic information can be displayed for a driver with respect to the observed context or as the reaction to contextual changes occurred; combined utilization of active context awareness (e.g., in critical situations) and passive context awareness (e.g., when user decision is required));
- Critical situation management (e.g., protocol-based collaboration, i.e., when agents recognize critical traffic situations, they can use corresponding predefined action plans for effective prevention/avoidance of these situations).

IV. The solution Based on Global Understanding Environment

The solution for UbiRoad challenges is based on results of SmartResource [8] and UBIWARE [10] projects. Their objectives were research and development of the large-scale environment for integration of smart devices, web services and humans based on Semantic Web and agent technologies. The projects belong to the Industrial Ontologies Group [17] research roadmap towards the Global Understanding Environment (GUN) [11, 12]. When applying Semantic Web in the domains of ubiquitous computing and smart spaces, it should be obvious that Semantic Web has to be able to describe resources not only as passive functional or non-functional entities, but also to describe their behavior (proactivity, communication, and coordination). In this sense, the word “global” in GUN has a double meaning. First, it implies that resources are able to communicate and cooperate globally, i.e., across the whole organization and beyond. Second, it implies a “global understanding”. This means that a resource A can understand all of (1) the properties and the state of a resource B, (2) the potential and actual behaviors of B, and (3) the business processes, in which A and B, and maybe other resources, are jointly involved.

According to GUN, resources (e.g., devices, humans, software components, etc.) can be linked to the Semantic Web-based environment via adapters (or interfaces), which include (if necessary) sensors with digital output, data structuring (e.g., XML) and semantic adapter components (e.g., XML to RDF). Agents are assumed to be assigned to each resource and are able to monitor data coming from the adapter about states of the resource, decide if more deep diagnostics of the state is needed, discover other agents in the environment, which represent “decision makers” and exchange information (agent-to-agent communication with semantically enriched content language) to get diagnoses and decide if any maintenance action is needed. Implementation of agent technologies and Multi-Agent Systems (MAS) within GUN framework allows mobility of service components between various platforms, decentralized service discovery, FIPA communication protocols utilization, and MAS-like integration/composition of services.

Agent-based layer of GUN-based architectures (e.g., of UbiRoad middleware), in addition to the agents, which are the representatives of the resources of interest, includes also an agent managing the *repository of roles and scenarios* encoded in RDF-based Semantic Agent Programming Language (S-APL) [13], an agent managing the *repository of reusable atomic behaviors* (i.e., software components that agents can load if a scenario prescribes), and an agent managing the *directory* that facilitates flexible discovery of agents (and thus of corresponding resources). S-APL – is a hybrid of semantics (metadata/ontologies/rules) specification languages, semantic reasoners, and agent programming languages. It integrates the semantic description of domain resources with the semantic prescription of individual and collaborative agents' behaviors.

A. *Universal Adapters for UbiRoad*

Semantic adapters layer is one of the most important layers of GUN and UbiRoad architecture (see Figure 2). Ideally the adapter should be that kind of software that is able to automatically reconfigure itself for each new resource based on its declarative description. As a result of adaptation any parameters observed, measured or collected elsewhere about the resource will be available in the same semantically rich format (RDF-based) referring some shared ontology. We developed RscDF (Resource State/Condition Description Framework) as a subset of S-APL and an appropriate format for adapters output [14]. It extends RDF by making it more suitable for semantic annotation of dynamic and context-sensitive data about the resources. It provides opportunity to put any RDF statement into context, which is described by a container of RDF statements. Appropriate schema also includes some specific properties able to describe dynamic and if needed multilayered context of statements. In [20] it is argued that there must be at least four categories of ontologies to represent and capture context-sensitive sensor data (devices ontology – to recognize different devices in the environment; context ontology – to model environmental information; data ontology – to make uniform of data coming from different sensors; and domain ontology – to represent a specific domain, e.g., collaborative traffic).

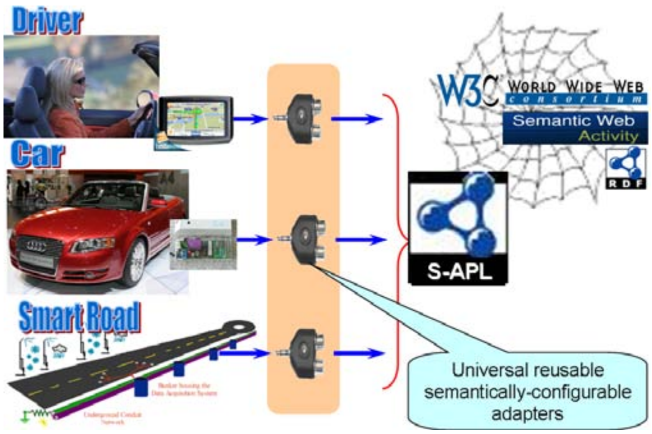

Figure 2. Semantic adapters for heterogeneous resources

B. *Reusable Behaviors for UbiRoad Components*

Behavioral layer is another important layer of GUN and UbiRoad and it is designed to make every domain resource proactive, which means able to autonomously behave towards achieving certain goals depending on its role in the domain. Such behavior depends on the nature and the type of the resource, its placement in the environment, relations with other resources, environmental parameters, etc. In UbiRoad, autonomy and proactivity of resources were implemented by means of software agents. The main challenge however was to avoid designing different agents for each of heterogeneous resources but implement just one universal behavior engine for an agent (to make it like an "artist"), which will be able to play any declaratively described behavior according to its current role. We require designing such reusable declarative behavior descriptions to be made with as minimal effort as possible and with maximal reuse of previously designed behaviors and their components when designing new ones (see Figure 3). Ideally the agent should be that kind of software that is able to automatically reconfigure itself for each new resource based on declarative description of this resource role in the domain or within some business process. We designed RgbDF (Resource Goal/Behavior Description Framework) as S-APL subset and a tool for semantic annotation of behavioral properties of the resource (goals, plans, roles, actions, intensions, etc.). It extends RDF by making it more suitable for semantic annotation of data about proactive and autonomous behavior of the resources [15]. The extension (in addition to the features provided by RDF, OWL and relevant reasoners) allows making explicit links from behavioral properties of proactive resources to appropriate atomic software components, which are intended to implement described behavior when appropriate. The roles (i.e., appropriate behaviors) of agents can be chosen and changed depending on current context of the situation, and this means that each agent should be able to download from some shared place the description of a new role whenever needed. Taking into account that some situations may be time-critical to react and that available semantic reasoners are not always able to provide decisions in real time (see analysis made in [21]), the UbiRoad solution allows (when appropriate) combining semantic reasoning (e.g., automated online generation of the actions plan in S-APL and implementing it) with the plans compiled in advance and available as hard-written (e.g., in Java) reusable atomic behaviors [13].

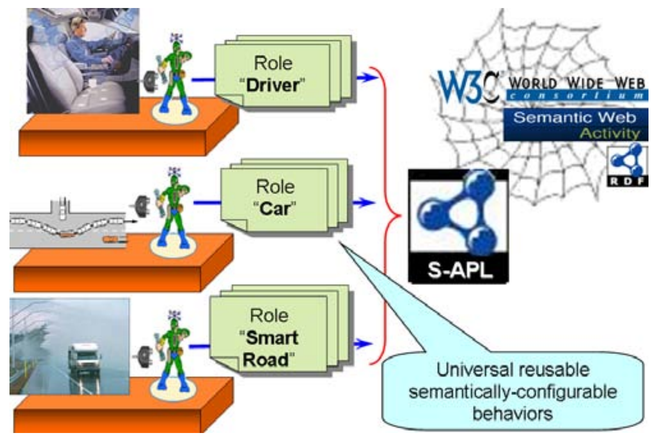

Figure 3. Reusable behaviors for UbiRoad "actors"

C. *Coordination of UbiRoad Resources*

The coordination layer is the next important layer of GUN and UbiRoad architecture and it is designed to make every domain resource collaborative, which means on the one hand coordination of autonomous and proactive parts of this resource (which are also smart resources themselves) and on the other hand coordinate own behavior with other resources within an organization (or within a scenario, which involves several individual proactive participants as shown in Figure 4) towards achieving consensus between personal and collaborative goals.

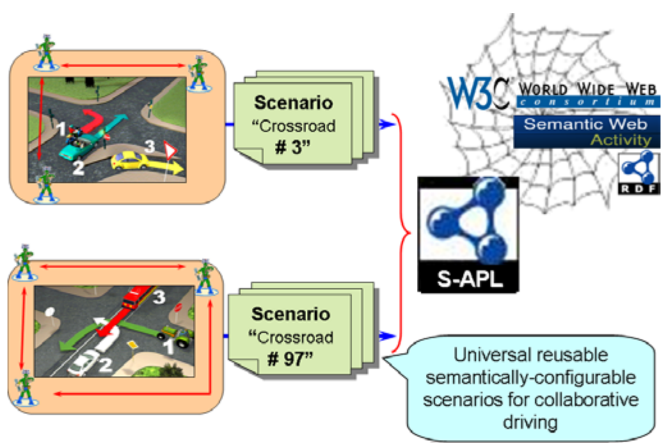

Figure 4. Reusable coordination scenarios in UbiRoad

We designed RpiDF (Resource Process/Integration Description Framework) as S-APL subset and a tool for semantic annotation of policies and metarules for controlling individual behaviors of the resources towards achieving collaborative goals. It extends RDF by making it more suitable for semantic annotation of collaborative behavior of the resources. The extension allows putting explicit constrains on individual rules, plans and utilized atomic behavioral software components, which are intended to implement corroborative goal-driven behaviors (scenarios) of the group of proactive resources (see Figure 4). It should provide ontologies and tools to design, share, reuse and integrate universal semantically-configurable scenarios for required coordination [16].

V. Utilization of the UBIWARE Platform's Features for UbiRoad

UBIWARE ("Smart Semantic Middleware for Ubiquitous Computing") has been developed by Industrial Ontologies Group (http://www.cs.jyu.fi/ai/OntoGroup) according to GUN vision. UBIWARE can be considered as a new software technology and a tool to support design and installation, autonomic operation and interoperability among complex, heterogeneous, open, dynamic and self-configurable distributed industrial systems, and to provide following services for system components: coordination, collaboration, interoperability, data and process integration. The UBIWARE project was a major step in a longer path that aims to build GUN (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 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 version UBIWARE 3.0 of the platform (Spring-Summer, 2010) appear to be a tool for creating and executing configurable distributed systems based on generalized and reusable business scenarios, which heterogeneous components (actors) are not predefined but can be selected, replaced and configured in runtime. Possible UbiRoad-related scenario on top of UBIWARE 3.0 platform is shown in Figure 5.

Extended version UBIWARE 3.1 (Summer-Fall 2010) is based on Cloud Computing architecture and provides both ontology-driven component- and scenario-based application design and configuration environment for the end-users and also platform-as-a-service to enable continuous run of the applications.

Several innovative features, technologies and components of the UBIWARE platform are making it as an excellent tool to enable the UbiRoad vision and appropriate software implementation. Therefore we will provide more details about it within the following text.

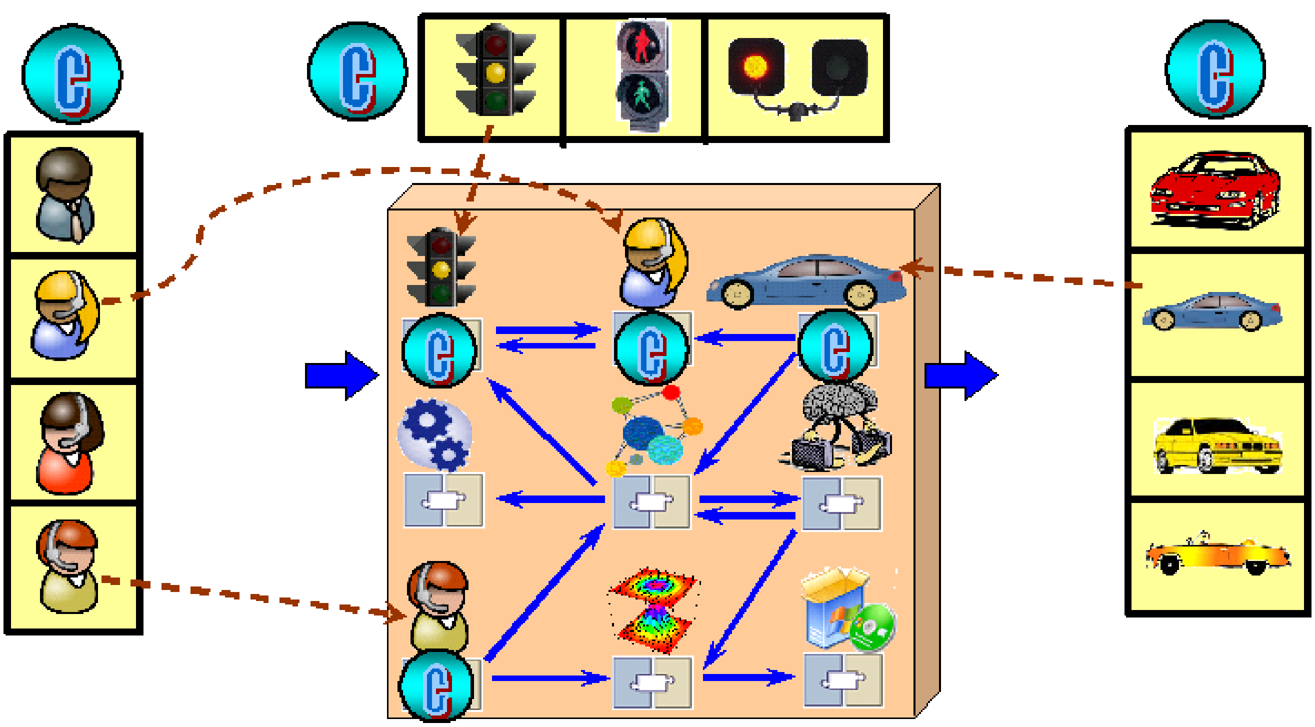

Figure 5. Abstract UbiRoad scenario implemented as a (self)configurable system on top of UBIWARE 3.0 platform

A. *OntoNuts – Proactive Semantic Adapters*

OntoNuts [22] have been proposed as an ontology-based instrument to enhance complex distributed systems by automated discovery and linking external sources of heterogeneous and dynamic data and capabilities during system runtime.

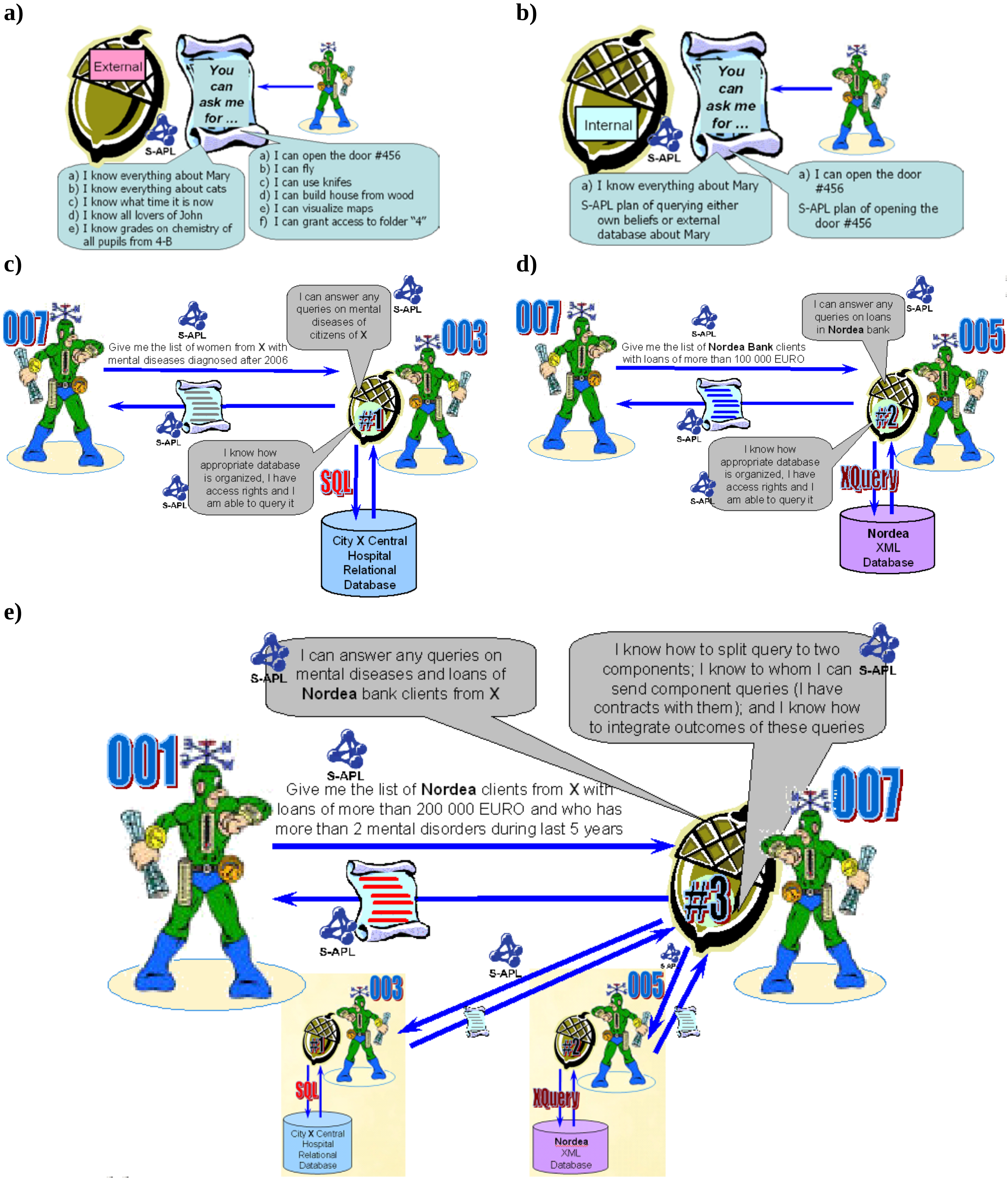

Figure 6. External (a) and internal (b) view on an OntoNut and examples of OntoNuts usage (atomic (c) and (d) and composite (e) OntoNuts)

An OntoNut can be seen from the two points of view and therefore it has two basic components: external and internal. External view to an OntoNut sees it as a tool for proactive advertising of capabilities of some external information (Figure 6 (a)) or service (Figure 6 (b)) provider. By such way each agent is able to actively advertise its capabilities (especially ones related to utilization of external services and databases) to other agents at the platform. Such advertisements generally include semantically annotated capability profile (presented in S-APL). Internal view to an OntoNut contains semantic description (S-APL) of the capability utilization plan for the agent with all needed information on how to access, invoke and monitor process of querying, executing and integrating of some external information sources or services.

Figure 6 (c) (also d) shows example of the OntoNut, which has been designed to wrap the capability of some external database querying. Due to such OntoNut the agent who has direct access to the database and knows how to make appropriate queries to get information from it, now will be also able to advertise such capability to other agents and provide appropriate service for them.

Interesting case is shown in Figure 6 (e) where complex OntoNut wraps two other OntoNuts and therefore is able to perform complex distributed query to two remote and heterogeneous (!) databases.

OntoNuts can be either preprogrammed by system designers or automatically created by agents themselves. Possible general rule for an agent of automatic OntoNut appearance can be presented like below:

IF I have the plan how to perform certain complex or simple action or the plan how to answer complex or simple query…

AND {time-to-time execution of the plan is part of my duty according to my role (commitment) **OR** I am often asked by others to execute action or query according to this plan}…

THEN I will create ONTONUT which will make my competence on this plan explicit and visible to others

OntoNuts approach looks similar to the Semantic Web Services [18] however provides much more potential due to agent-driven proactivity of services (or service components) and data sources. Added value of proactivity for similar purposes has been described in [19].

B. *4i (For-Eye) – Visualization-as-a-Service*

4i (For-Eye) [23-24] is a smart ontology-based visualization technology able to automatically discover and utilize external visualization service providers and dynamically create and visualize mashups from external data sources in a context-driven way.

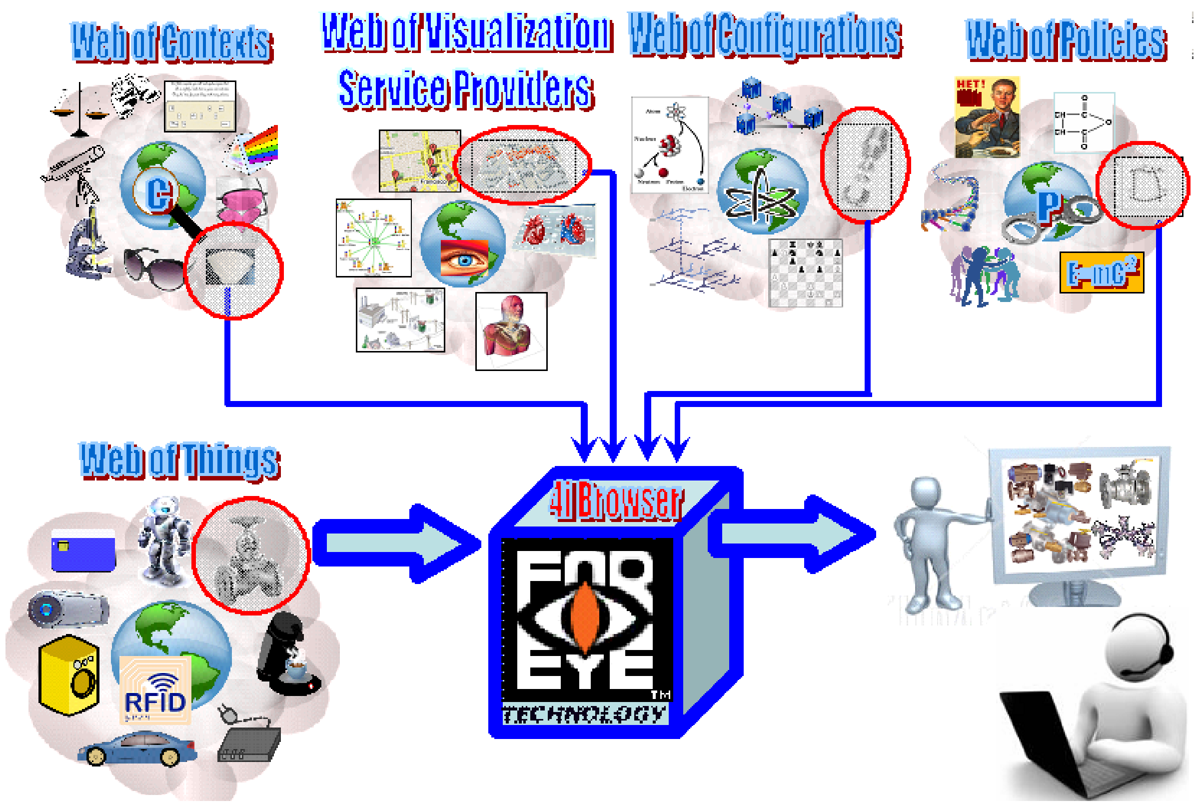

Figure 7. For-Eye-Browser illustrated: Visualization-as-a-Service applied to ubiquitous objects

While OntoNuts are used for advertising, remote access and utilization of external capabilities (software or ubiquitous), 4i technology is applied to enable "Human-as-a-Service" by providing both: interface from UBIWARE infrastructure to a human and vice versa. The key of 4i approach is that instead of designing human interfaces (which is quite labor-consuming task) for each needed combination of data sources, presentation, context, etc, it is proposed to utilize (reuse and integrate when appropriate) external interfacing software components acting as visualization service providers. Therefore slogan of 4i technology is "Visualization-as-a-Service".

The generic vision of future tool (4i Web Browser), restricted version of which is currently part of UBIWARE platform, is shown in Figure 7. Given URI of some resource (document, device, human, etc.) and the task is to visualize it. The selection of properties (to be shown) of the resource as well as the neighborhood of it depend on the context of the concrete viewer. Browser should be aware about the context (either by explicit provision of it from the user or by referencing to URI of some annotated reusable context published in the Web). Also the configuration of the resource may be explicitly provided similarly to the context; and the policy applied to current visualization (which components or properties of the resource can be shown to such a user with particular access rights and which not). Finally appropriate visualization service will be discovered and utilized, which is able to show such kind of resources and their neighborhood on the screen (Figure 7).

C. *Smart Comments – User-Driven Configuration Tool*

Smart Comments – is smart ontology-based technology for end-user-driven control and configuration management of the application in runtime based on smart mapping of appropriate tags from natural language comments provided by a SW engineer and the source code.

Via Smart Comments an end user (not a programmer) will be able to modify the business logic of an application. Figure 8 illustrates such possibility. If S-APL programmer at the design phase leaves a Smart Comment (natural language description synchronized with the code through several configurable variables) attached to some S-APL construct (e.g. S-APL rule as shown at the picture), then the interface for this rule modification can be automatically generated and called during runtime by the end-user. S-APL modifications can be applied immediately to the constantly running system without stopping it.

Therefore Smart Comments are serving both for documenting S-APL code (supported by UBIWARE engine) and for automatic generation of end-user interface for system reconfiguration whenever needed on the fly.

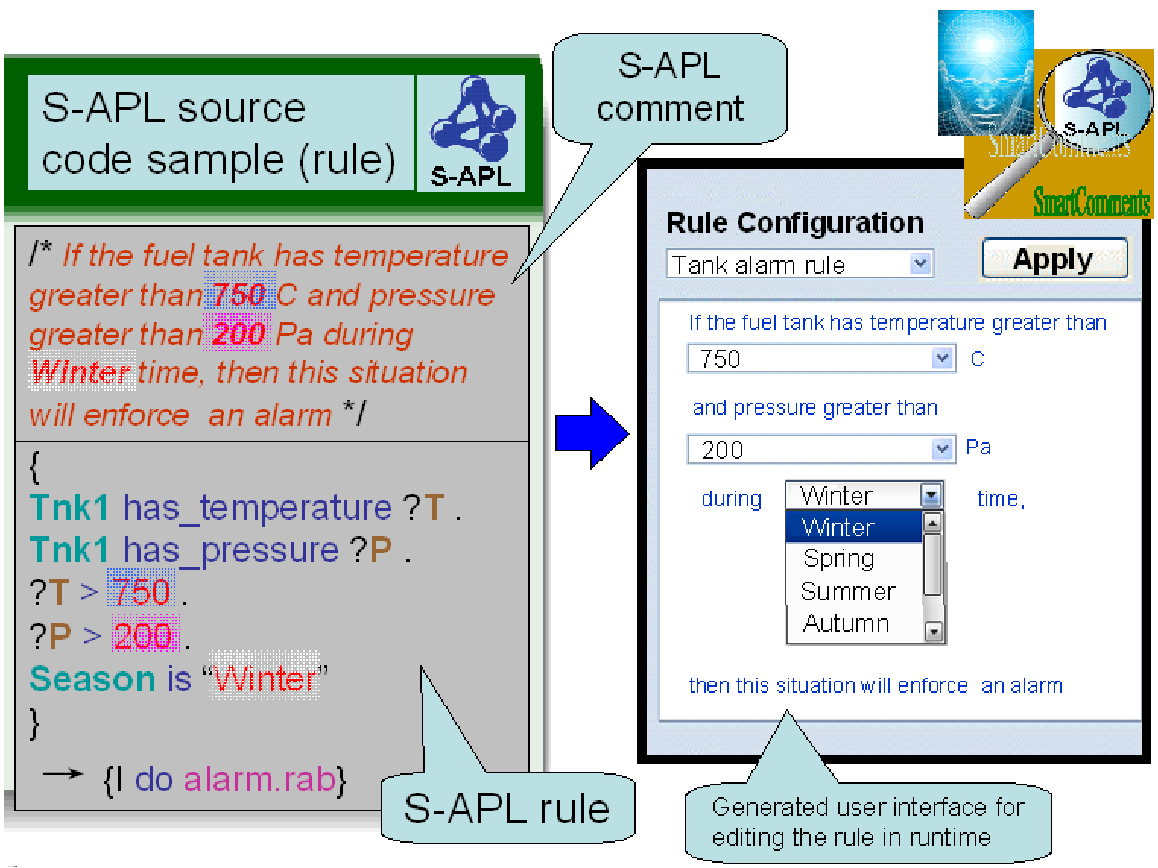

Figure 8. Example of a Smart Comment and appropriate reconfiguration user interface

D. Semantic Blogging – Collecting Annotated History

Blogging or collecting and publishing own history (usually by humans) is very popular feature of e.g. various Web 2.0 applications. Taking into account that UBIWARE and especially UbiRoad applications are supporting complex distributed business processes, which involve various components (humans, vehicles, devices, services, infrastructure, etc.), we assume that blogging may serve not only to humans but to other ubiquitous or software entities. Own history (especially semantically annotated one) collected by each system component can be later processed by various intelligent tools and useful patterns can be discovered and reused.

In Figure 9, the semantic blogging is shown for the maintenance lifecycle management of a vehicle. Integrated information from sensors, fault detectors, diagnostic software or humans, maintenance workers, etc., collected in timely fashion and automatically annotated provides valuable source of data for predictive maintenance of that particular vehicle and possibly of the same type of vehicles. In UBIWARE, semantic blogging is agent-driven and all history data is represented in S-APL (i.e. can be processed and exchanged by agents).

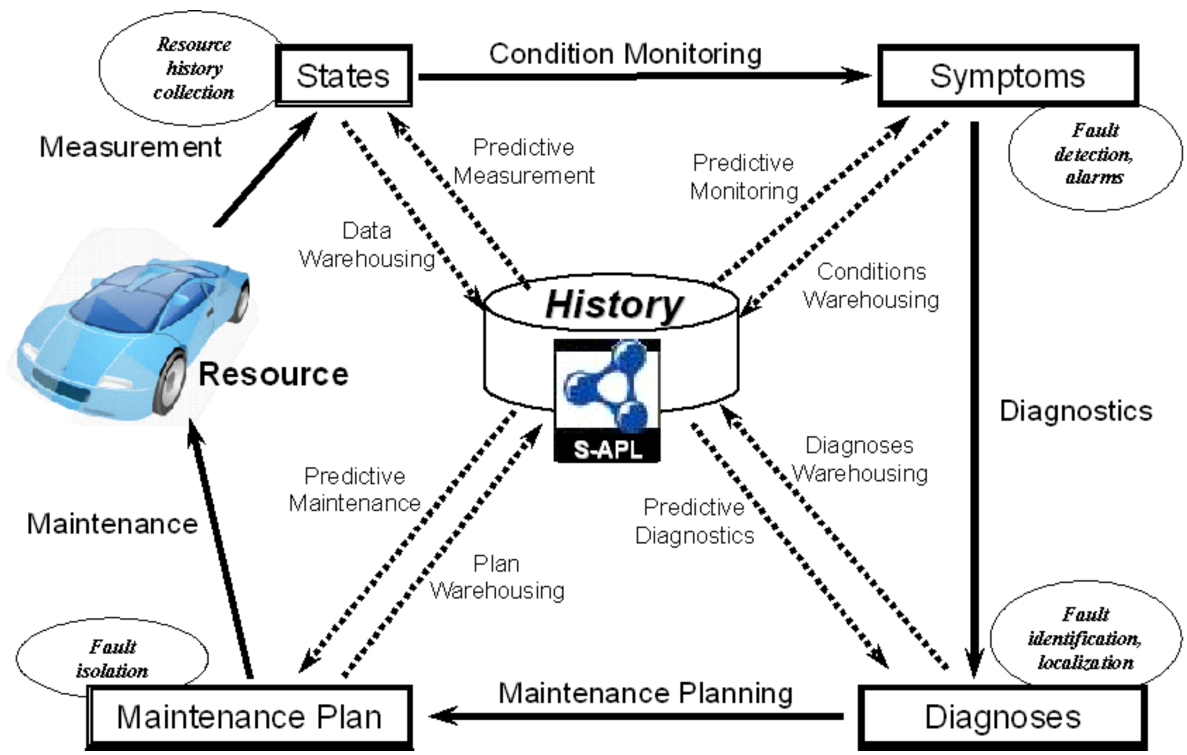

Figure 9. Lifecycle of a semantic blog related to vehicle maintenance domain

VI. UBIWARE: Innovative Software Architecture

The UBIWARE Platform is a development framework for creating complex self-managed multi-agent systems in various domains (not only for the UbiRoad). It is built on the top of the Java Agent Development Framework (JADE), 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, a multi-agent system is seen, first of all, as a middleware providing interoperability of heterogeneous 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 10 together with four main innovations behind it (approach, engine, language, OntoNuts). 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 (see updates in [17]) contains a set of RABs and the libraries that simplify UBIWARE application development.

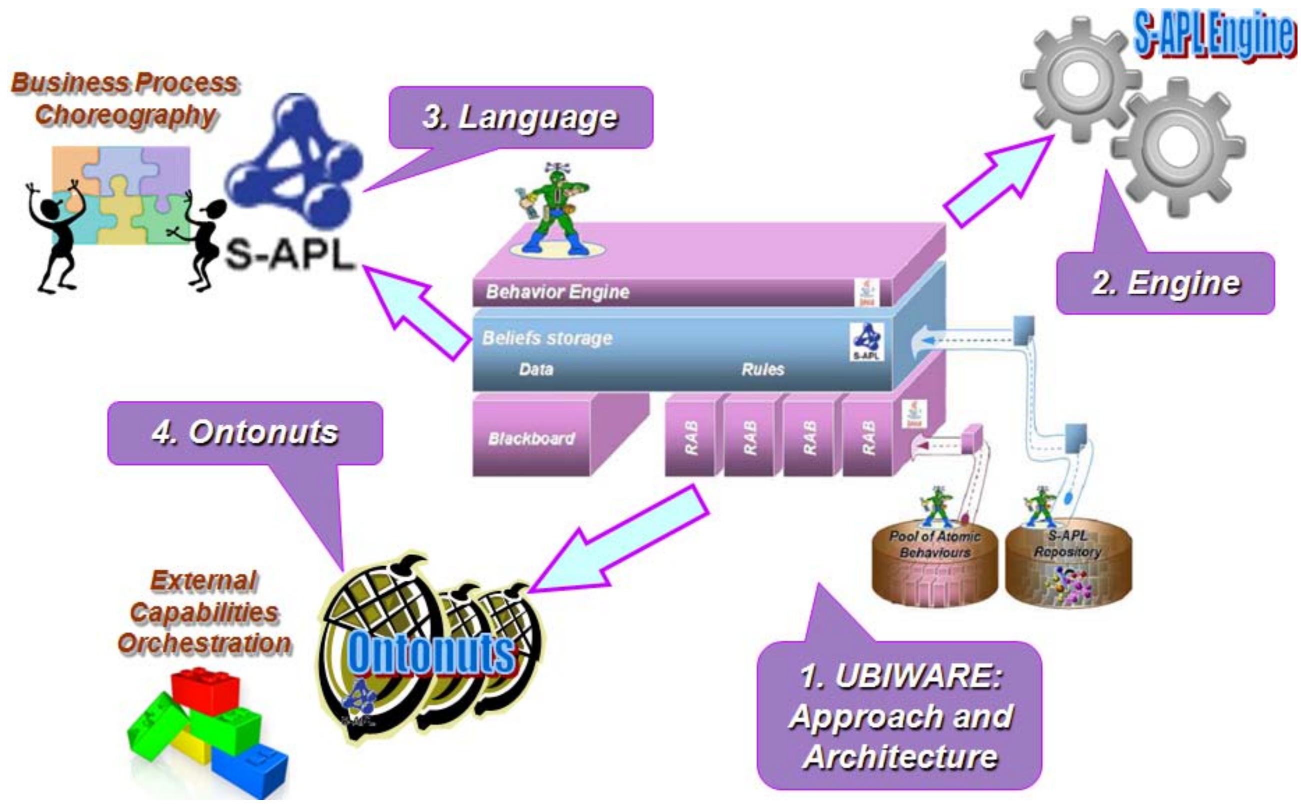

Figure 10. UBIWARE agent architecture with four main innovations

The middle layer is the beliefs storage in 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), business process description languages (like BPEL) and agent communication content languages (like FIPA SL). 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 advantage is that in S-APL 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.

Together with the main UBIWARE engine, an OntoNuts technology and OntoNuts engine have been implemented. OntoNuts technology tackles the problem of distributed querying in UBIWARE-based multi-agent systems. Due to it and based on the Open World Assumption (alternatively to e.g. BPEL with the Closed World Assumption), the OntoNuts engine is able to discover and automatically utilize (in a runtime) new services (capabilities) that have just appeared in semantic service registry or/and 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, a special type of OntoNut called DoNut has been implemented 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. It is known that the Service Oriented Architecture (SOA) is an approach of 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. UBIWARE platform nowadays is such a middleware solution that 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, the personal user agents have been introduced, thus making the platform user-oriented 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. Latest architecture of the platform follows cloud computing paradigm combined with agent architecture, in which two groups of agents were identified. The first group includes the agents which are application-specific, whereas the second group gathers infrastructure agents providing services to those application-specific ones.

VII. Ontology and Semantic Integration for Traffic Management

To enable interoperable scenarios on top of UBIWARE platform (written in S-APL) and seamless integration of external capabilities there is a need for shared domain ontology. To run UbiRoad application on UBIWARE, the Traffic & Mobility Ontology is being developed as collaborative effort of Industrial Ontology Group, VTT (Technical Research Center of Finland) and Cooperative Traffic ICT SHOK Consortium. Due to complexity of the domain and heterogeneity of components, standards and actors there, such effort is a quite challenging task which will include:

- Vehicles Ontology
- Drivers Ontology
- Infrastructure Ontology
- Logistics Ontology
- Organizations/Products/Services Ontology
- Behavioral Ontology
- Monitoring/Diagnostics/Control/Maintenance Ontology
- Cooperative Scenarios Ontology
- Policy Ontology (security, privacy, safety, economic, skills/demands, environmental, operational, institutional, personal, cultural, etc.)

Ontology is especially important for the OntoNuts technology performance because it allows externalizing not only data sources, services and other capabilities, but also remote and heterogeneous systems as whole.

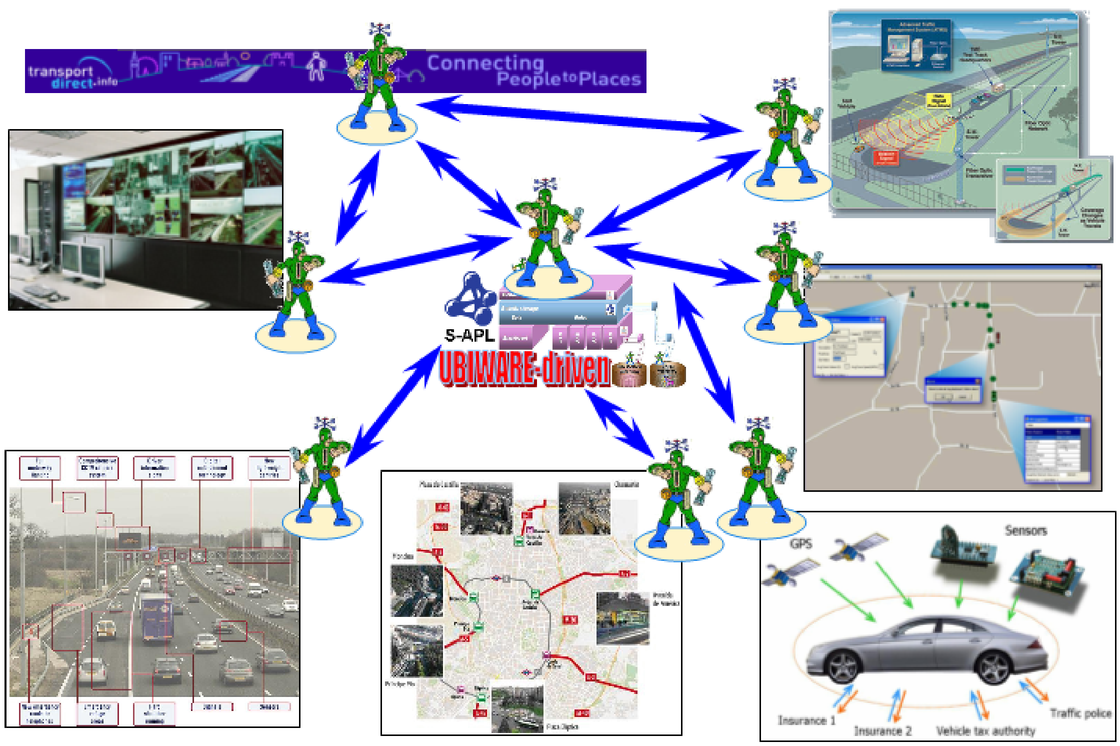

Figure 11. Agent-mediated heterogeneous traffic management systems integration

In Figure 11 it is shown that services provided by remote traffic management systems (heterogeneous, distributed, “self-interested”) can be automatically advertised, mediated, utilized and integrated via agent-driven OntoNuts. UBIWARE platform in this case will act as kind of smart semantic “glue” linking and integrating remote systems as services according to user-driven scenarios.

VIII. Related work

The topics related to cooperative traffic, traffic management systems, smart traffic environments, driver assistance, etc., are quite popular nowadays. It would be quite challenging to observe all various approaches and solutions in these fields. However not all of them recognize the needs and benefits, which semantic and agent technologies may bring to provide scalability, flexibility and interoperability for available solutions, systems and products. Lets observe few additional samples of research efforts, which are also basing their solutions on semantic or/and agent technology.

The Intelligent Systems Group's (ISG) (http://www.ee.oulu.fi/research/isg) is trying to develop enhanced adaptivity and context-awareness for smart environments. The research specifically focuses on the creation of dynamic models that enable monitoring, diagnostics, prediction and control of target systems (living and artificial) or operating environments making the environment adapt to the users, instead of making the users adapt to an environment. In [25] ISG present their work towards achieving context-awareness in mobile devices by combining Semantic Web technology with sensory data. They show that some context data pertaining to the user, such as location, time, and physical surroundings, is vital for the realization of intelligent maps able to reason on the sensory data with the help of appropriate ontologies and to utilize its inference output to achieve context-awareness.

Researchers from AI & CS Lab, University of Porto, in [26] explored potential benefits of concepts such as visual interactive modeling and simulation to implement a cooperative network editor embedded in a collaborative environment for transport analysis. They argue that traditional approaches lack adequate means to foster integrated analyses of transport systems either because they are strict in terms of purpose or because they do not allow multiple users to dynamically interact on the same description of a model. They show that the use of semantic approach and a common geographical data model of the application domain enable different experts to interact seamlessly in a collaborative environment.

Efforts of NEARCTIS (FP7-th Network of Excellence for Advanced Road Cooperative traffic management in the Information Society, http://www.nearctis.org/) focus on cooperative systems for road traffic optimization, and it covers a wider scope as it appears that cooperative systems have to be integrated into the whole traffic management systems. One of challenges they recognized is to provide means for sharing resources (data, experimental tools, bibliographical databases), organizing the spreading of the knowledge and research results, for which one cannot avoid utilization of Semantic Technologies.

Deducing spatial knowledge for car driver assistance is of special importance for emerging Advanced Driver Assistance Systems. Such systems cannot rely only on in-car sensors infrastructure, but also require thorough environmental tracking. In [27] an approach is presented of a distributed ad-hoc infrastructure that collects and disseminates tracking data of environmental objects and allows ontology-based reasoning. It is shown that such a system can facilitate driver assistance based on spatial knowledge.

In general, driver assistance system demands a common domain understanding for scene representation to enable information exchange between a vehicle and a driver. In [28] an ontology modeling approach is presented for assisting drivers through safety alerts during time critical situations. Designed Intelligent Driver Assistance System (I-DAS) manages appropriate alert parameter representation in XML format while the recognition and interpretation of a critical situation is done using ontology. Authors argue about the feasibility of combining the advantages of ontology with the reasoning power of logic-based languages.

Researchers from Advanced Highway Maintenance and Construction Technology Research Center, University of California-Davis, are investigating issues related to transportation asset management and related infrastructure maintenance. In [29] they present their asset management solution based on combination of semantic models of mobile and stationary transportation assets with the visualization capabilities of Google Earth. Semantic models can represent complex relationships between diverse asset classes and Google Earth is used for visualization because of its accessibility to a wide range of users and ability to combine different types of data. The model defines stationary and mobile assets, and real-time traffic sensors. Results show that the developed semantic models facilitate integration of appropriate software and hardware systems.

Vehicular traffic in modern cities makes our mobility there quite time and resource consuming. Therefore we expect from future traffic management systems significant savings of fuel and time if traffic control mechanism could be effectively discovered.

In [30] the problem of real time traffic data availability and processing is handled by utilizing ubiquitous database and intelligent agents for traffic data management. Ubiquitous database provides automatic everywhere access to the data and so the called unique routing agent is used to handle the distribution of the database, route discovery and maintenance. The method has been simulated for the measurement of traffic related parameters (traffic load, occupancy and trip time).

IX. CONCLUSIONS

In this paper, we approach the traffic-collaboration-support problem from the semantic viewpoint. In other words, the semantic technologies have a two-fold value in UbiRoad. First, they are the basis for the discovery of heterogeneous resources and data integration across multiple domains (a well-known advantage). Second, they are used for behavioral control and coordination of the agents representing those resources (a novel use). Therefore, semantic technologies are used both for descriptive specification of the services delivered by the resources and for prescriptive specification of the expected behavior of the resources as well as the integrated system (i.e., declarative semantic programming). While the standard semantic technology is capable of effective description of static resources only, the UbiRoad is a tool for semantic management of content relevant to dynamic, proactive, and cooperative resources. The agent technology is extended by developing tools for semantic declarative programming of the agents, for massive reuse of once generated or designed plans and scenarios, for agent coordination support based on explicit awareness of each other's actions and plans, and for enabling flexible re-configurable architectures for agents and their platforms applied for cooperative traffic domain. However, taking into account that the efficiency of semantic technologies and available tools for real-time applications as well as agent technologies (e.g., agent negotiation within fast developing situations) is still questionable, a smart way to combine semantic and agent approaches with efficient online data processing and automation tools would be reasonable.

This paper is an extended version of conference paper [1] accepted for journal publication.

REFERENCES

[1] Terziyan, V., Kaykova, O., and 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, pp. 295-302.

[2] Hype Cycle for Automotive Integration and Communication Technologies, 2006, Gartner.

[3] Chen-Ritzo, C-H, C. Harrison, J. Paraszczak, and F. Parr. "Instrumenting the Planet." IBM Journal of Research & Development. Vol. 53. No. 3. Paper 1. 2009.

[4] Cooperative Traffic ICT: Strategic Research Agenda, May 2009, ICT SHOK, Finland, Available in: http://www.cooperativetraffic.fi/ images / 1 /16/SRA_CT_v2.pdf .

[5] Buckley, J., From RFID to the Internet of Things: Pervasive Networked Systems, Final Report on the Conference organized by DG Information Society and Media, Networks and Communication Technologies, Brussels, 2006. Available in: http://www.rfidconsultation.eu/docs/ficheiros/WS_1_Final_report_27 _Mar.pdf .

[6] Kaykova, O., Khriyenko, O., Kovtun, D., Naumenko, A., Terziyan, V., and Zharko, A., General Adaption Framework: Enabling Interoperability for Industrial Web Resources, International Journal on Semantic Web and Information Systems, 1(3), 2005, pp. 31-63.

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

[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, pp. 33-57.

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

[10] Katasonov, A., Kaykova, O., Khriyenko, O., Nikitin, S., and 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, pp. 169-178.

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

[12] Terziyan, V. and 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, pp. 55-87 (Chapter III).

[13] Katasonov, A. and Terziyan, V., Implementing Agent-Based Middleware for the Semantic Web, In: Proceedings of the International Workshop on Middleware for the Semantic Web in conjunction with the Second IEEE International Conference on Semantic Computing (ICSC-2008), August 4-7, 2008, Santa Clara, CA, USA, IEEE CS Press.

[14] Kaykova, O., Khriyenko, O., Naumenko, A., Terziyan, V., and Zharko, A., RSCDF: A Dynamic and Context-Sensitive Metadata Description Framework for Industrial Resources, Eastern-European Journal of Enterprise Technologies, 3(2), 2005, pp. 55-78.

[15] Kaykova, O., Khriyenko, O., Terziyan, V., and Zharko, A., RGBDF: Resource Goal and Behaviour Description Framework, In: M. Bramer and V. Terziyan (Eds.): Industrial Applications of Semantic Web, Proceedings of the 1-st International IFIP/WG12.5 Working Conference IASW-2005, August 25-27, 2005, Jyvaskyla, Finland, Springer, IFIP, pp. 83-99.

[16] Katasonov, A. and Terziyan, V., Semantic Approach to Dynamic Coordination in Autonomous Systems, In: R. Calinescu et al. (Eds.), Proceedings of the Fifth International Conference on Autonomic and Autonomous Systems (ICAS 2009), April 21-25, 2009, Valencia, Spain, IEEE CS Press, pp. 321-329.

[17] Web Pages of the Industrial Ontologies Group: Available in: http://www.cs.jyu.fi/ai/OntoGroup/index.html .

[18] Arroyo, S., Lara, R., Gomez, J., Berka, D., Ding, Y., and Fensel, D., Semantic Aspects of Web Services: Practical Handbook of Internet Computing, Chapman & Hall and CRC Press, USA, 2004, 31.31-31.17.

[19] Ermolayev, V., Keberle, N., Plaksin, S., Kononenko, O., and Terziyan, V., Towards a Framework for Agent-Enabled Semantic Web Service Composition, International Journal of Web Service Research, Idea Group, 1(3) , 2004, pp. 63-87.

[20] Bell, D., Heravi, B., and Lycett, M., Sensory Semantic User Interfaces (SenSUI), In: Proceedings of the International Workshop on Semantic Sensor Networks, 2009, pp. 96-109.

[21] Luther, M., Liebig, T., Bohm, S., and Noppens, O., Who the Heck is the Father of Bob?, In: Proceedings of ESWC-2009, Springer, LNCS 5554, pp. 66-80.

[22] Nikitin, S., Katasonov, A., and Terziyan, V., Ontonuts: Reusable Semantic Components for Multi-Agent Systems, In: R. Calinescu et al. (Eds.), Proceedings of the Fifth International Conference on Autonomic and Autonomous Systems (ICAS 2009), April 21-25, 2009, Valencia, Spain, IEEE CS Press, pp. 200-207.

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

[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), Barcelona, Spain, 28-31 July, 2007, pp. 233-240.

[25] Su, X., Riekki, J., and Tarkoma, S., An Approach to Achieve Context-Aware Maps: Combining Semantic Web Technology with Sensor Data, In: Proceedings of the 5th International Conference on Intelligent Environments (IE09), Barcelona, Spain, 2009, pp. 193-203.

[26] Pereira, J., Rossetti, R., and Oliveira, E., Towards a Cooperative Traffic Network Editor, In: J. Luo (ed.), Proceedings of the 6th International Conference on Cooperative Design, Visualization, and Engineering (CDVE'09), LNCS, Vol. 5738, Springer, 2009, Luxembourg, pp. 236-239.

[27] Tonnis, M., Klinker, G., and Fischer, J., Ontology-Based Pervasive Spatial Knowledge for Car Driver Assistance, In: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07), NY, USA, 2007, pp. 401-406.

[28] Kannan, S., Thangavelu, A., and Kalivaradhan, R., An Intelligent Driver Assistance System (I-DAS) for Vehicle Safety Modeling using Ontology Approach, In: International Journal of UbiComp, Vol.1, No.3, July 2010, pp. 15-29.

[29] Darter, M., Lasky, T., and Ravani, B., Transportation Asset Management and Visualization Using Semantic Models and Google Earth, In: Transportation Research Record: Journal of the Transportation Research Board, Vol. 2024, 2008, pp. 27-34.

[30] Dave, N., and Vaghela, V., Vehicular Traffic Control: A Ubiquitous Computing Approach, In: Communications in Computer and Information Science, Springer, 2009, Vol.40, Part 7, pp. 336-348.