

A Hybrid Technology for Operational Decision Support in Pervasive Environments

Alexander Smirnov, Tatiana Levashova, Nikolay Shilov, Alexey Kashevnik

St.Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS), 39, 14th line, St.Petersburg, 199178, Russia
{smir, oleg, nick, alexey}@ias.spb.su

Abstract The paper addresses the issue of development of a technology for operational decision support in a pervasive environment. The technology is built around the idea of using Web-services for self-organization of heterogeneous resources of the environment for decision support purposes. The approach focuses on three types of resources to be organized: information, problem-solving, and acting. The final purpose of the resource self-organization is to form an ad-hoc collaborative environment, members of which cooperate with the aim to serve the current needs according to the decision situation. The hybrid technology proposed in the paper integrates technologies of ontology management, context management, constraint satisfaction, Web-services, and intelligent agents. The application of the technology is illustrated by response to a traffic accident.

1 Introduction

Pervasive environment consists of increasing number of heterogeneous resources communicating through interconnected network. These resources produce large volumes of information and provide different services. The goal of pervasive computing is to enable computing anywhere at anytime. This goal can be achieved through spontaneous organization of the surrounding resources in a context aware manner to meet the real-time needs.

The paper proposes a hybrid technology intended for operational decision support in a pervasive environment. It combines technologies of ontology management, context management, constraint satisfaction, Web-services, and intelligent agents. The technology sustains context-sensitive self-organization of the surrounding resources according to a decision situation (a situation in which decisions are to be made). The purpose of the resource self-organization is to form an ad-hoc resource collaborative environment, members of which cooperate with the aim to serve the current needs of the operational decision support.

The idea behind is to use Web-services as mediators between the pervasive environment and the surrounding resources. It is proposed to represent the resources by sets of Web-services. The set of Web-services representing each resource implements the functionality of this resource. This makes it possible to replace the self-organization of resources with that between the Web-services. In terms of this replacement the resource collaborative environment is correspond to an ad-hoc service network.

The decision situation is modeled at two levels: abstract and operational. At the abstract level the decision situation is represented by the *abstract context* that is an ontology-based model of this situation expressed by constraints. At the operational level the decision situation is represented by the *operational context* that is an instantiated abstract context. The operational context is produced by the self-organized service network representing resources to be collaborated.

The decision support system (DSS) built upon the hybrid technology is based on service-oriented architecture. The architecture enables interaction with the heterogeneous resources using the ad-hoc Web-service network and Web-service communications using an agent-based service model [1].

2 Hybrid Technology

The hybrid technology sustaining operational decision support in pervasive environments is based on ontology engineering, ontology management, context management, constraint satisfaction, profiling, Web Services, and intelligent agents. These technologies are applied depending on the objectives the DSS meets at the particular moment (Table 1).

Effects and advantages of exploiting ontology in DSSs are universally recognized now [2]. The approach offered here uses application ontology (AO) for representation of the knowledge of the application domain. The AO is created by subject experts, knowledge and ontology engineers. It either can be created from scratch or through integration of existing ontologies. In the former case *ontology engineering* technology is used. In the latter case ontology engineering technology is integrated with *ontology management*. The AO is specified by means of the formalism of object-oriented constraint networks (OOCN) [3] in order that tasks specified in this ontology can be solved as constraint satisfaction problem (CSP).

Resources are represented by sets of *Web-services*. To provide the Web-services with semantics [4] the Web-service descriptions are aligned against the AO. Alignment is an objective of *ontology engineering* and *ontology management*.

The technologies mentioned up to this point are involved at a preliminary stage preceded to the stage of decision support. The stage of decision support starts with introducing the decision situation to the DSS through the user (decision maker) request. The request specifies the type of the situation to be modeled.

Table 1. Technological framework

Objectives	Techniques	Technology	Result in terms of OOCN
AO building	Integration of existing ontologies, knowledge formalisation	Ontology engineering, ontology management	OOCN without variable values
Representation of resource functionalities	Alignment of the AO and Web-service descriptions	Ontology engineering, ontology management, Web-services	OOCN without variable values
Extraction and integration of relevant knowledge	Abstract context composition	Ontology management	General problem model
Self-organization of Web-services	Agent interactions	Intelligent agents	Instantiated problem model
Gathering and processing of relevant information	Operational context producing	Context management, Web-services	Instantiated problem model
Search for a solution	Problem solving	Constraint satisfaction, Web-services	A set of feasible solutions
User preferences revealing	Context-based accumulation of made decisions	Profiling	A set of user constraints

Based on the type of the situation the DSS extracts knowledge relevant to this type from the AO and integrates it in the *abstract context* that is an ontology-based model of the situation. The knowledge is extracted along with Web-services, descriptions of which are aligned against this knowledge. To operate on the extraction of knowledge and its integration *ontology management* methods are applied.

The abstract context is the base for self-organization of Web-services that have been included in the abstract context, into a service network. The purpose of the service network is the organization of a resource collaborative environment for producing an *operational context* and for taking joint actions required in the situation. The operational context is the instantiated abstract context or an instantiated model of the decision situation. The operational context is interpreted as CSP by the service network using the *constraint satisfaction technology*.

Self-organization of the Web-services is carried out through negotiation of their needs and possibilities. To make the Web-services active components capable to self-organize an agent-based service model is used. *Intelligent agents* negotiate services' needs and possibilities in terms of the AO negotiating input (service needs) and output (service possibilities) arguments of the functions that the Web-services implement. To operate on the producing the operational context technologies of *context management*, *Web-services*, and *intelligent agents* are involved.

The decision situation (the operational context) and a set of solutions for tasks represented in this context are presented to the decision maker. The solution chosen by the decision maker is considered to be the decision. The abstract and operational contexts, the set of solutions, and the decision are saved. The DSS uses them for revealing user preferences. This is the focus of the *profiling* technology.

3 Service-Oriented Architecture

In the architecture (Fig. 1) of the DSS intended for functioning in a pervasive environment two types of Web-services are distinguished: *core Web-services* and *operational Web-services*.

The core Web-services are intended to support the DSS users and the abstract context creation. These Web-services comprise:

- **UserProfileService** creates, modifies, and updates the user profile; provides access to this profile; collects information about the user; accumulates information about the made decisions in a context-based way; reveals user preferences;
- **UserInteractionService** is responsible for interactions of the DSS with its users. It communicates between the DSS and its users providing DSS messages, context-sensitive help, pictures of decision situations, results of problem solving, and delivering information from the users to the DSS;
- **AOAccessService** provides access to the AO;
- **AbstractContextService** creates, stores, and reuses abstract contexts;
- **ManagementService** manages Web-services to create the abstract context. It operates with the service registry where the core services are registered.

The operational Web-services self-organize a Web-service network. In the ar-

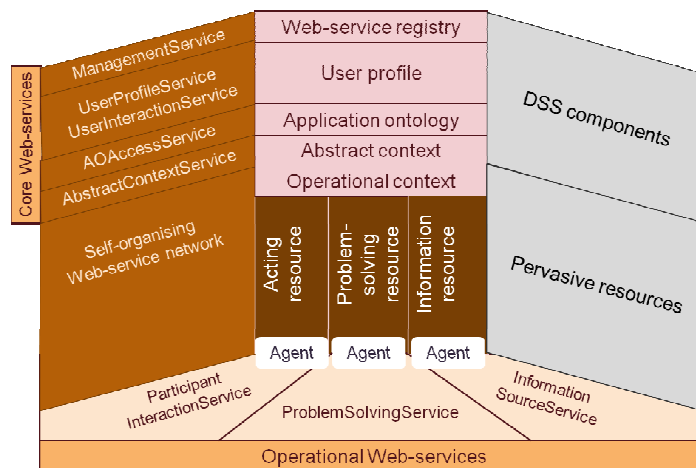


Fig. 1. Service-oriented architecture

chitecture the information, problem-solving, and acting resources are distinguished. The information resources are resources providing information to the abstract context from the information sources. The problem-solving resources are computational modules, applications, services, *etc.* that can be used to solve problems formalized in the abstract context. The acting resources are organizations and persons acting in decision situations according to their roles.

The set of operational Web-services comprises:

- **InformationSourceService** – a set of Web-services responsible for interactions with information sources of different types and for processing information provided by these sources. The following main types of information sources are distinguished: *sensors, databases, Web-sites, and humans*;
- **ProblemSolvingService** – a set of Web-services in charge of problem solving.
- **ParticipantInteractionService** that is a set of Web-services responsible for support of and interactions with acting resources. They provide communication between the DSS and human representatives of the acting resources.

4 Case Study: Traffic Accident

The hybrid technology is illustrated by decision support to response to a traffic accident that was caused by the ignition in the car petrol tank. The AO used for this purpose represents knowledge of the emergency management domain.

As soon as the DSS receives the information that a traffic accident had happened, the DSS creates or reuses the abstract context characterizing the accident situation. An exemplified piece of the abstract context (the taxonomy) for the traffic accident is shown in Fig. 2. Because of the petrol tank ignition the context combines knowledge required to response to both traffic accident and fire event.

Within the abstract context the acting resources fall into classes *Actor* and *Job Role*. The *emergency medical service organisation* is responsible for providing emergency teams (*emergency medical technicians* in Fig. 2), *ambulances*, and *rescue helicopters* for emergency medical care of injured people and / or for transportation them to *hospitals*. The *fire department* is responsible for providing firefighter brigades (*firefighters* in Fig. 2), *fire trucks*, and *fire helicopters* for fire extinguishing. *Local police organisation* is responsible for providing *police officers* and *police trucks* to investigate the accident and to go through formalities. Two types of transportation are possible: *Air transportation* used by fire and rescue helicopters and *Automobile transportation* used by ambulances, fire trucks, and police trucks.

Problem solving knowledge is collapsed in the class *Emergency Response*. This class formalizes tasks to be solved to respond to the traffic accident. They are as follows.

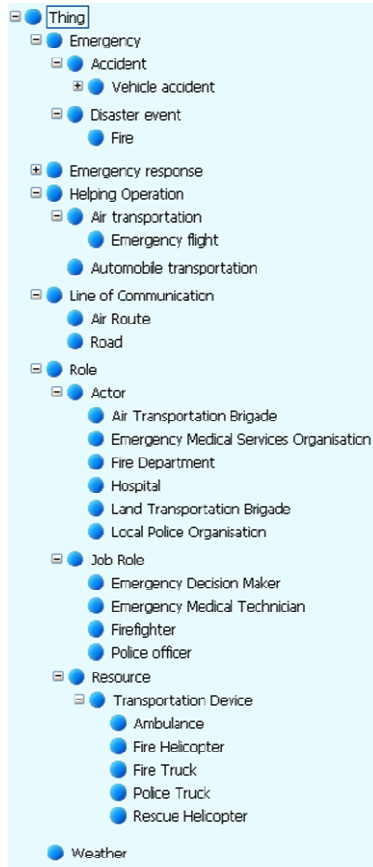
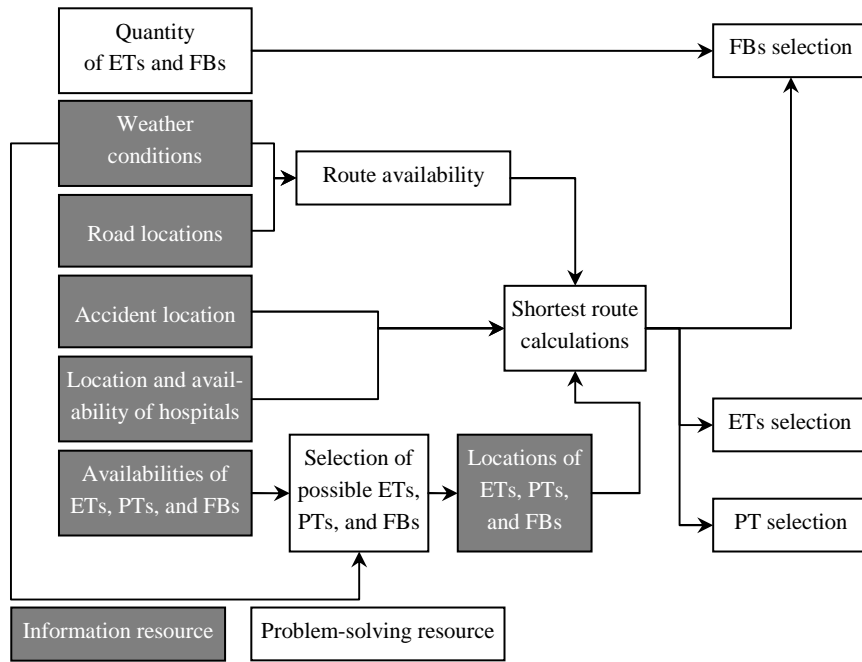


Fig. 2. Abstract context: taxonomy

- The task *Quantity of emergency teams and firefighter brigades* calculates the required quantity of these kinds of groups.
- The task *Brigade availability* determines the availability of emergency and traffic police teams, and firefighter brigades.
- The task *Brigade location* determines the current location of emergency and traffic police teams, and firefighter brigades.
- The task *Hospital availability* returns a list of hospitals of the region, hospital addresses, free capacities, and hospital availabilities.
- The *Route availability* task determines availability of a particular route depending on its type (road, air route, etc.) taking into account (i) the types of vehicles used by the emergency and police teams, and firefighter brigades; (ii) the closed roads; (iii) the traffic jams; and (iv) the weather conditions.
- The *Shortest routes* task calculates the shortest routes for the appropriate acting resources.
- Joint solution for the tasks *Firefighter brigade selection*, *Emergency team selection*, *Hospital selection*, *Police team selection*, and *Route selection* produces a set of feasible plans of actions for the acting resources.

The network of Web-services organized to solve the tasks described above is shown in Fig. 3. Arrows in the figure depict execution sequences of the Web-services. The tasks implemented by Web-services organizing parallel paths can be solved simultaneously.

Web-services implementing tasks concerned with supplying the DSS with the data from the information sources use the following kinds of information resources. The current weather conditions are taken from the *sensors* and *Web-sites*. Information about the locations of the roads of the region is taken from the *GIS*. Information about emergency teams, firefighter brigades, and police teams available in the region is read from a *database*. Information about the locations of these teams and brigades is provided by the *GPS-based devices* installed on the vehicles of these teams and brigades. Information about the accident location, its type (traffic accident with fire), and the approximate number of victims is provided by the car *smart sensor*. Information about hospitals available in the region and their lo-



ET – Emergency Team; PT – Police Team; FB – Firefighter Brigade

Fig. 3. Network of information and problem-solving Web-services

cations is read from the healthcare infrastructure *database*, hospital free capacities are provided by *hospital administration systems*.

The common model of a Web-service implementing information resource functions is illustrated by the example of Web-service responsible for receiving information about emergency and police teams, and firefighter brigades available in the region. This Web-service requests the database storing information about the emergency and police teams, and firefighter brigades and returns a list of such teams and brigades. The list contains identifiers of the teams and brigades, URIs of their Web-services (this Web-services are used to receive additional information about the teams and brigades, e.g. current brigade location), and types of vehicles used by these teams and brigades. The Web-service being illustrated is implemented in PHP [5]. Key steps of the service are as follows:

```

    ...
    $conn = odbc_connect ("brigades"); //connection to the database storing
    information about the emergency and police teams, and firefighter
    brigades
    
```

```

$sql = "SELECT id, brigade_Description, brigade_WebserviceURI,
brigade_Type, brigade_WorkType FROM brigades"; //query to the
database
// id - identifier of the brigade
// brigade_WebserviceURI - URI of the Web service of the brigade
// brigade_Type - type of the vehicle the brigade uses
// brigade_WorkType - type of the brigade (emergency team or
firefighter brigade or police team)
$brigades = GetData($conn, $sql); //query result returned to the Web
service in OOCN compatible format
// GetData is responsible for the conversion of information from
database format into OOCN compatible format
...
return $brigades; //Web service output - a list of brigades with
their characteristics

```

Problem solving resources are invoked in a similar way as it is illustrated for the information resources.

The traffic accident scene (operational context) and the set of action plans are presented to the decision maker. The decision maker chooses one solution (Fig. 4) from the generated set that is to be the decision. The solution presented in the figure is generated for 4 traffic accident victims, and 8 emergency teams, 8 firefighter brigades, and 3 police teams found in the region. Dotted lines in Fig. 4 depict routes to be used for transportations of the selected teams and brigades.

The decision is delivered to the leaders of the emergency teams, firefighter brigades, police teams, and to the hospital administrations. They have access to the operational context through any Internet browsers (a browser supported by a notebook, PDA, mobile phone, *etc.*).

The DSS has been implemented as a distributed system for operational decision support. The interface of the system is Web-based, i.e. regular Web browsers can be used for working with the system.

5 Related Research

The field of pervasive computing is a focus of many up-to-date research efforts. Research in this area aims at management of sensor networks, industry of context-aware services including self-adaptable, self-configurable, self-optimized services, development of resource-aware services, building smart spaces, *etc.*

Organization of collaborative environments from autonomous entities is a focus of approaches to building context aware DSSs [e.g., 5, 7], self-optimization and self-configuration in wireless networks [8], organization of context-aware cooperative networks [9] and collaborative context-aware service platforms [10], *etc.* Issues close to the subjects investigated in the research presented in this paper are considered in the framework of PLASTIC (Providing Lightweight & Adaptable Service Technology for Pervasive Information & Communication) [11] project sponsored by the EC FP6. Among other issues in this project a service oriented

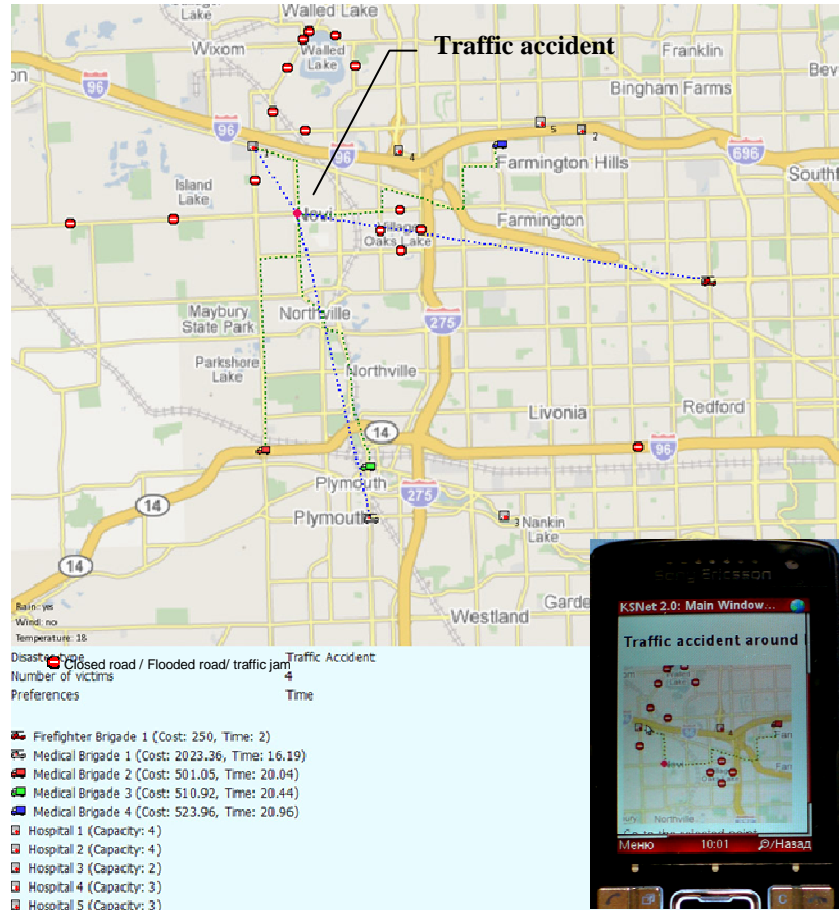


Fig. 4. Operational context: plan for actions

approach allowing for ad hoc context awareness in pervasive environments has been developed. In the PLASTIC-approach context-related entities – all taking the form of contextual services – are purposed to dynamically self-organize and self-adapt to optimally exploit available, possibly heterogeneous, contextual resources at the specific time and place.

The approach described in this paper combines the ideas of context aware decision support with the ideas of self-contextualization and self-organization. Distinguishing features of the proposed approach comparatively to the related ones are using Web-services to ensure interoperability between heterogeneous devices of pervasive environment, involving problem-solving resources in customization of the environment functionality, and organization of a collaborative environment comprising not only physical devices and software services but organizations and persons.

6 Conclusion

The paper proposes a hybrid technology that enables to self-organize resources of the pervasive environment to collaborate for the purposes of operational decision support. It is shown that self-organization of physical resources can be substituted for self-organization of Web-services representing these resources. To make the Web-services able to self-organize, an agent-based service model can be used. Agents make Web-services “active” components. Interactions of Web-services can be supported by formal interface agreement defined by the technology of Web-services enriched with ontology semantics.

Acknowledgements The paper is due to the research carried out as a part of the project funded by grant 08-07-00264 of the Russian Foundation for Basic Research, and project 213 of the research program “Intelligent information technologies, mathematical modelling, system analysis and automation” of the Russian Academy of Sciences.

References

1. Hao Q, Shen W, Wang L (2006) Collaborative Manufacturing Resource Scheduling Using Agent-Based Web Services, *Int J Manuf Technol and Manag*, 9(3/4):309-327.
2. Saremi A, Esmaeili M, Habibi J, Ghaffari A (2008) O2DSS: A Framework for Ontology-Based Decision Support Systems in Pervasive Computing Environment. *Modeling & Simulation, Proceedings 2nd Asia International Conference (AICMS 08)*, 41-45.
3. Smirnov A, Levashova T, Pashkin M, Shilov N, Komarova A (2007) Disaster response based on production network management tasks. *Management Research News*, Emerald Publishing Group. 30(11):829-842.
4. Bandara A, Payne T, De Roure D et al (2008) A Pragmatic Approach for the Semantic Description and Matching of Pervasive Resources. *Proceedings 3rd Int Conf on Grid and Pervasive Computing (GPC 2008)*. 434-446.
5. The PHP Group: “PHP” (2009). <http://www.php.net>. Accessed 15 January 2009
6. Kwon O, Yoo K, Suh E (2005) UbiDSS: a Proactive Intelligent Decision Support System as an Expert System Deploying Ubiquitous Computing Technologies. *Expert Syst Appl* 28(1):149-161.
7. Burstein F, Zaslavsky A, Arora N (2005) Context-aware mobile agents for decision-making support in healthcare emergency applications. In: Bui T, Gachet A (eds). *Proceedings Workshop on Contextual Modelling and Decision Support*, Paris, France. CEUR Workshop Proceedings, 144: http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-144/07_burstein.pdf. Accessed 15 January 2009.
8. Self-optimisation and self-configuration in wireless networks (SOCRATES) (2008). ICT-2007.1.1 The network of the future. <http://cordis.europa.eu/>. Accessed 14 January 2009.
9. Ambient Networks Project (2006). IST-2004-2.4.5 Mobile and Wireless Systems and Platforms Beyond 3G. <http://www.ambient-networks.org/>. Accessed 14 January 2009.
10. Ejigu D, Scuturici M, Brunie L (2008) Hybrid Approach to Collaborative Context-Aware Service Platform for Pervasive Computing. *J Computers*. 16(1):40-50.
11. PLASTIC. <http://www.ist-plastic.org/> (2008). Accessed 11 January 2009.