Web-Service Network for Disaster Management

Alexander Smirnov, Nikolay Shilov, Tatiana Levashova, Alexey Kashevnik
St.Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences,
St.Petersburg, Russia
{smir, nick, oleg, alexey}@iias.spb.su

ABSTRACT

The paper addresses the issue of context-aware operational decision support in emergency situations. A decision support system (DSS) developed for this purpose is implemented as a network of a set of Web-services. The Web-services try to organise a service network according to context. Here the context is proposed to be modelled as a "problem model". It specifies problems to be solved in a particular kind of emergency situation. Context is produced based on the knowledge extracted from the application domain (application ontology) and formalised by a set of constraints. The purpose of the service network is provision the DSS with contextualised information from diverse information sources and solving problems specified in the context. In the framework of context-aware operational decision support, composition of the application ontology for the disaster management domain from the Semantic Web Ontologies is discussed and a hybrid technology of context-aware operational decision support is presented. The technology is based on ontology management, context management, constraint satisfaction, and Web Services. Application of the ideas above is illustrated by an example of a decision support system for real-time resource coordination and situation awareness for logistics management in fire response operations.

Keywords

Disaster response, constraint satisfaction, context, self-organising network, operational decision support, Web-services.

INTRODUCTION

Crises are an increasingly common feature of modern life. September 11, Indian Ocean tsunami, Hurricanes Katrina and Rita are just a few examples of crisis events happened recently. Crisis response operations involve multiple actors that collaborate to carry out tasks on crisis response. The actors have to collaborate in fast-changing environments; information about crisis development is collected from a variety of human and non-human sources. In the dynamic setting of a crisis event, it is very difficult to keep an overview of all the ongoing activities and information flow within the entire environment. Different actors have different tasks and roles to fulfil. Actors are not fully aware of what is happening and are not aware of what they might need to know (Van Someren, Netten, Evers, Cramer, Hoog and Bruinsma, 2005). Therefore the task of efficient coordination of activities of organisation and human resources that may participate in a crisis response operation is an important issue.

The research presented in the paper addresses the task of efficient coordination of activities of multiple resources proposing self-organising network of Web-services. The purpose of organisation of the service network is to integrate contextual information from diverse information sources and to provide the participants of the emergency response operation with situation awareness and with plans of action. Resources to be coordinated comprise organisational, human, and computational ones. Web-services are responsible for interactions with these kinds of resources and management of them.

For representation of an emergency situation context model is used. A context is the basis for organisation of Web-services in a service network. The Web-services communicate within the context using a common vocabulary of the dosaster domain. In the research context is proposed to be modelled as a "problem model". It is produced based on the knowledge extracted from the application domain and formalised by a set of constraints. The purpose of the context is twofold. It is aimed at making knowledge and information relevant to the emergency situation sharable by the resources and to supplying a decision support system (DSS) to information provided by these resources.

The problem is suggested being modelled by two types of contexts: *abstract* and *operational* (Smirnov, Pashkin, Shilov and Levashova, 2007). The problem formalised by the set of constraints can be processed as *constraint* *satisfaction problem* (CSP). Solving problems specified in the context is a purpose of the service network. The result of CSP solving is one or more satisfactory solutions for the problem modelled.

Knowledge of the disaster management domain is represented by an application ontology. To come to an agreement between ontology-based problem representation and constraint-based problem solving formalism of object-oriented constraint networks for ontology representation is used.

A hybrid technology proposed for context-aware operational decision support is based on ontology management, context management, constraint satisfaction, and Web Services.

The paper summarizes the results obtained during the theoretical research and its embodiment in the disaster management domain. Application of the ideas discussed in the paper is illustrated by an example of a DSS scenario for real-time resource coordination and situation awareness for logistics management in fire response.

The remainder of the paper in organised as follows. The next Section outlines related research. Then, the conceptual framework of context-aware operational decision support is described and the procedure of application ontology composition is discussed. Next, a network of Web-services organised for fire response purposes is presented and a scenario of multi-role decision support within this network is considered. At last, this scenario is illustrated in the context of real-time resource coordination. Some concluding remarks are presented in Conclusion.

RELATED WORK

Decision support in emergency situations is a focus of many studies. State of affairs, problems and needs of emergency management are described in (Harrald, Jefferson, Fiedrich, and Sener, 2007).

Some approaches dealing with decision support for emergency response concentrate on the problem of collaborative decision making. Among them an approach addressing the problem of real time decision making for groups working in a changing environment (Gouman, Kempen, de Vree, Capello, van der Heijden and Wijngaards, 2007). A system implementing this approach is intended to support collaborative work of a large group of professionals over the duration of a disaster event. It collects information about the current state of emergency situation. The information is accessible by a large group of experts from all over the world. The methodology behind the approach proposes methods for coordination of individual decisions to come to an organized and efficient response. Collaborative decision making under chaotic circumstances is a focus of empirical research in the context of a crisis management exercise (White, Turoff, and Bartel Van de Walle, 2007).

The EU-IST project SHARE (Löffler, Hernández, Schon, Pottebaum and Koch 2007) aims at developing a mobile service architecture to support large-scale rescue operations with multimedia communication and information services. SHARE technology is intended to support rescue teams working on large-scale rescue operations by providing GIS enabled decision support. In the focus of applications offered by SHARE are decision support, accessibility of crucial operation information and collaborative work through synchronized information management as well as communication with improved efficiency. Within the SHARE's objectives the problem of management of material and human resources depending on human roles and qualifications is considered.

Semantic technologies, automated web mining and human language technologies are integrated (Chapman and Ciravegna, 2006; Potter, Kalfoglou, Alani, Bachler, Buckingham Shum, Carvalho, Chakravarthy, Chalmers, Chapman, Hu, Preece, Shadbolt, Tate and Tuffield, 2007) to provide focused mining of unstructured data to create a timely structured data repository. The e-Response system using these technologies collates information in real time to aid responding to emergencies. It automatically extracts information from web material referring to the geographical area desired. Using an incremental approach a knowledge base for the immediate area surrounding the emergency is built. The whole system is centred around an ontology which defines the disaster domain.

Formal ontological analysis of threat is proposed to enhance decision makers' abilities to make informed decisions (Little and Rogova, 2006). The analysis is based on a formal ontological structure of threats as integrated wholes possessing three inter-related parts: intentions, capabilities and opportunities.

The mentioned research has much in common with the presented here research, especially as regards personalised user support, information processing, and dynamic resource management. However, these approaches mainly are aimed at collecting and processing of information relevant to the current emergency situation and presentation this information to the user as a base for making decisions, i.e. provision the users with situational pictures. The DSS described in the paper provides decision makers not only with a situational picture but supports them with solutions for tasks to be solved in particular kinds of emergency situation.

CONCEPTUAL FRAMEWORK

This section describes the developed methodology for the context-aware operational decision support for disaster management and technology supporting the set of methods and models proposed within the methodology.

Methodology

The main idea behind the methodology framework consists of (i) creation of an ontology-based model of the problem to be solved by the user (decision makers, and other participants involved in the decision making process) or of the situation to be presented to the user and (ii) solving the problem as CSP. A situation is considered as the problem model where the data values change over time.

An ontology-based model of the problem is composed of knowledge extracted from an application ontology (AO). The AO represents knowledge of an application domain that is disaster management here. The AO does not hold instances. Instead, it refers to Web-services responsible for supplying the DSS to data values. Harmonisation of Web-service descriptions and the AO is a time-consuming procedure. Because of this, they are harmonised in advance, at the stage of AO creation. This allows the DSS to reduce time taken at the stage of decision making.

The Web-services obtain data values as results of i) interactions with information sources from which data values can be taken and ii) problem solving. The procedure of the AO creation is supported by the subject experts, knowledge and ontology engineers. The process of composition of this ontology is described in Section "Application Ontology".

For internal ontology representation the formalism of object-oriented constraint networks (OOCN) (Smirnov, Pashkin, Shilov, and Levashova, 2007) is used. The reason of this is that application of constraint networks allows simplifying the formulation and interpretation of real-world problems that are usually defined as constraint satisfaction problems in such areas as management, planning, etc. The formalism of OOCN provides mechanisms for declarative problem modelling. Constraints are considered as special predicates of first-order logic. The problem model represented by means of this formalism can be interpreted as CSP. CSP model consists of three parts: a set of variables; a set of possible values for each variable (its domain); and a set of constraints restricting the values that the variables can simultaneously take. The result of its solving is a set of feasible solutions that satisfy all the constraints specified in the OOCN.

The starting point for knowledge extraction is a type of disaster that the user enters in his / her request to the DSS. Knowledge relevant to this type of disaster is extracted from the AO. This knowledge is integrated into the *abstract context* that is an ontology-based problem model without values for model variables. This context is saved for possible further reuse. The abstract context is the basis for self-organisation of Web-services in a network. The Web-services, the knowledge included in the abstract context refers to, organise themselves in a network aimed at solving the problems specified in this context. Solving the problems implies obtaining data values for model variables and search of a set of feasible solutions for the problem modelled, i.e. problem solving itself.

When the network of the Web-services has been organized the Web-services start instantiating the abstract context. The instantiated abstract context is an *operational context* that is the problem model along with problem data and at the same time is an OOCN to be interpreted as a CSP. The operational context along with options for problem solving is presented to the user for making decisions. Different tasking, competencies, and responsibilities of multiple users involved in the operation on disaster response are taken into account as user roles. Picture of the operational context presented to a user is restricted to the information that is useful to the user filling a particular role.

Referring to the disaster management domain, the abstract context, for instance, may represent transport facilities for people evacuation. Whereas, the operational context gives an explicit idea about conditions (number of victims, weather conditions, state of the roads, etc.) in the disaster region, based on which a decision about facilities to be used is made.

Hybrid Technology

Context-aware operational decision support is a complex compound problem. Its treatment involves actions on acquisition and representation of domain knowledge, processing of dynamic information flows, producing a real-time picture of the current situation, problem solving, and coordination of decisions made by different actors. These actions are purposed by different technologies. Moreover, up-to date DSSs are required to be able to functioning in distributed environments. To go through all the actions a hybrid technology that unites existing intelligent technologies was proposed (Smirnov, Pashkin, Shilov, and Levashova, 2007) and described in full.

The hybrid technology is based on *profiling*, *ontology management*, *context management*, *constraint satisfaction*, and *Web Services*. *Profiling* techniques are used for organization of personalised user support. *Ontology management* techniques are used to operate on the capturing and integration of relevant knowledge. *Context management* methods are applied for modelling needs of the user, i.e. what problems are to be solved in the current situation and what information needed for this. *Constraint satisfaction* technology is used for problem solving. The *Web-services* are used to exploit distributed data and reasoning resources. They are responsible for providing services according to the DSS needs.

APPLICATION ONTOLOGY

According to the formalism of OOCN the AO is represented with a set of *classes*; a set of class *attributes*; a set of attribute *domains*; and a set of *constraints*. The set of constraints comprises 1) (*class, attribute, domain*) relationship used to model triple of classes, attributes pertinent to them, and restrictions on the attribute value ranges; 2) *taxonomical* ("is-a") and *hierarchical* ("part-of") relationships used to model class taxonomy and class hierarchy respectively; 3) classes *compatibility* used to model condition if two or more instances can be parts of the same class; 4) *associative* relationships used to model any relationships besides the listed ones; 5) class *cardinality* restriction used to define how many subclasses the class can have; 6) *functional* relations used to model functional dependencies between the class attributes. The formalism of OOCN is supported by a WebDESO tool developed for ontology management purposes (Smirnov, Pashkin, Chilov, and Levashova, 2003).

The AO being created is purposed to represent conceptual knowledge used to model different disaster events, infrastructures, locales, weather conditions, resources and their roles, etc. and to formalise a set of problems that may require solutions during different disaster response operations. Since construction of such an ontology from scratch demands significant expert efforts and a lot of ontologies have been published so far, it has been decided to create the AO through reuse of existing ontologies.

The AO is made up of two constituents: domain knowledge and task knowledge. Correspondingly, the process of composition of the AO involves formalisation of the both kinds of knowledge.

Domain knowledge for the disaster management domain was composed from Semantic Web Ontologies, access to which the Swoogle[1] engine provides for. Ontologies containing knowledge relating to the domain in question were found in the following sources: Guavahbsont[2], GEOFILE[3], CyC[4], The Component Library[5], and Ontosem[6].

The source format of the found ontologies was OWL Web Ontology Language. Using special mechanisms (Smirnov, Paskin, Chilov Levashova and Krizhanovsky, 2003) these ontologies were converted into the formalism of OOCN. After that parts of these ontologies were integrated in a single ontology by the experts. In the ontology some modifications were introduced. Mainly, they concern specification of hierarchical, associative, class cardinality, and class compatibility constraints.

Task knowledge was formalised completely by the experts. The formalism of OOCN represents tasks as classes, sets of their arguments and arguments' types are represented by sets of attributes and domains, respectively. The tasks are configured in accordance with task-subtask decomposition structure, where tasks and their subtasks are related by "part-of" relationships.

Domain and task knowledge within the AO are interrelated with functional constraints. These constraints (i) designate instances of the classes of the domain knowledge and (ii) state which domain knowledge is involved in task solving and in what way.

---

[1] http://swoogle.umbc.edu/index.php?option=com_frontpage&service=search&queryType=search_swd_ontology-&searchString=&searchStart=1

[2] http://ontologies.isx.com/onts/uav/guavahbsont.owl, Ontology catalogue of ISX Lab of Lockheed Martin Advanced Technology Labs

[3] http://www.daml.org/2001/02/geofile/geofile-ont, DAML ontology library

[4] http://semweb.mcdonaldbradley.com/.dev/OWL/Cyc, Library of The University of Maryland, Baltimore County

[5] http://www.cs.utexas.edu/~mfkb/RKF/tree/CLib-core-office.owl, Knowledge Systems Group, UT-Austin

[6] http://morpheus.cs.umbc.edu/aks1/ontosem.owl, Library of The University of Maryland, Baltimore County

In the paper a list of tasks to be solved by the DSS for real-time resource coordination during fire-related response operations is given. In reality, tasks represented in the AO are not restricted by the fire event and this exemplified list. The list of tasks formulated by the experts is as follows:

- define the number of emergency teams and firefighter brigades taking into account type of disaster and quantity of victims;
- determine route availabilities based on the current weather conditions;
- select emergency teams for taking injured people to hospitals based on the current team locations, team availabilities, hospital locations and availabilities, and route availabilities;
- select firefighter brigades for extinguishing fire and means for their transportations based on the current brigade locations and availabilities, and route availabilities;
- design a plan of actions for the emergency teams and firefighter brigades.

The task knowledge of the AO has been formalised as shown in Figure 1. The task *Quantity of emergency teams and firefighter brigades* calculates the required quantity of these kinds of groups for the relief operation. The tasks *Brigade availability* and *Brigade location* determine the availability and the location of both emergency teams and firefighter brigades. The task *Hospital availability* returns a list of hospitals of the region, hospital addresses, free capacities, and hospital availabilities.

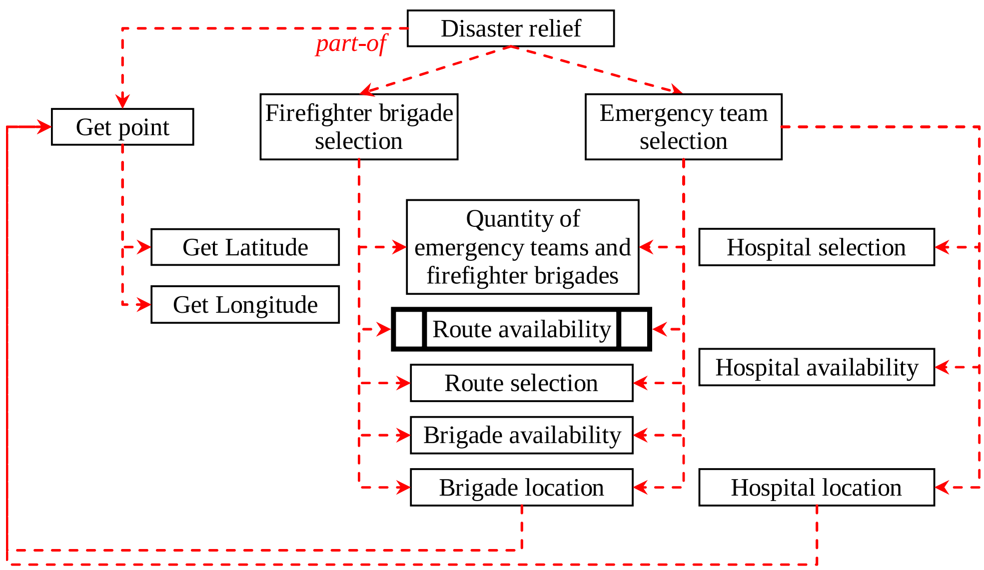

Figure 1. Application ontology: class view for task knowledge

Tasks concerning determination of any locations (locations of brigades, hospitals, routes, etc.) return these locations using a GIS (Geographic Information System). The GIS is used as the information source for the tasks *Get point*, *Get Latitude*, *Get Longitude*. The *Get point* task returns an array of points of the region with their coordinates. Correspondingly, the tasks *Get Latitude* and *Get Longitude* return the coordinates of a given point.

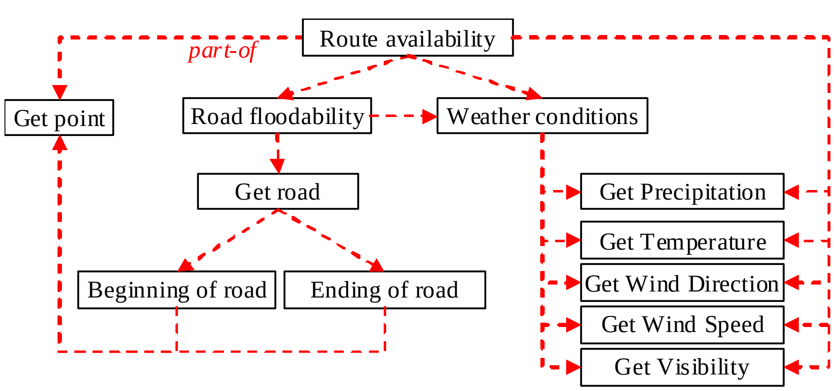

Figure 2. Route availability task

The task of determination of the availability of a route is expanded in Figure 2. The task *Get Road* using GIS returns an array of the roads for the current region. The task *Road floodability* checks either a particular road is floodable or not under the current weather conditions. The current weather conditions are returned as results of solving the task *Weather conditions*.

This task integrates data provided by various resources focusing on weathercasting. The *Route availability* task determines availability of a particular route depending on its type (road, air route, etc.) taking into account the weather conditions.

The tasks have been described so far are intended to be solved before the user is presented with the operational context. As soon as these tasks were solved the operational context has received values for all the variables represented in it. In other words, all the attributes occurring in the abstract context have been instantiated.

Unlike the tasks above, the tasks *Firefighter brigade selection*, *Emergency team selection*, *Hospital selection*, and *Route selection* require a decision. The expected decision is about which emergency teams are chosen to transport injured people to which hospitals, which firefighter brigades are chosen to go to the fire location, and which routes to be used for the transportation. Joint solution for these four tasks produces a plan of actions for the emergency teams and firefighter brigades.

Web-service referencing

In the AO references to Web-services are specified as follows. For AO classes that represent tasks, URIs of Web-services responsible for these tasks are introduced. It does not matter if a Web-service obtains data values from information sources or solves a problem. Then from the list of functions this service implements a function corresponding to the task in question is selected. At last, correspondences between the function's input and output arguments and the task's arguments in the AO are indicated. The procedure of Web-service referencing is supported by the ontology engineers with help of the WebDESO tool.

Results of task solving serve as input arguments for super-tasks or / and as values for instantiation of AO classes representing the domain knowledge. In the AO, the fact that an attribute takes a value that is a result of a Web-service function is specified through introducing a functional constraint. This constraint is introduced between the class attribute and an output argument of the task. This argument specifies the value the Web-service function returns as the task solution.

The proposed ontology-based referencing to Web-services allows the Web-services to exchange information about their needs and possibilities in terms on the ontology vocabulary. Web-services describe their needs through a set of input arguments of the functions that these Web-services implement. Correspondingly, the possibilities are described through a set of output arguments of the functions. Relationships (functional constraints) between the arguments and attributes of the classes representing the domain knowledge enable the Web-services to communicate about their arguments in terms of the domain knowledge.

ABSTRACT CONTEXT

The abstract context is produced for the user entered fire as the type of disaster in his / her request to the DSS. According to the procedure of abstract context producing the names for classes and attributes of the AO are checked for their matching with the word "fire". Classes and attributes whose names match with this word serve as input for the ontology slicing operation. The purpose of this operation is to capture ontology abstractions surrounding the found classes and attributes so as the captured knowledge would be relevant to the user request and to the inference supported by OOCN. The result of the operation is a set of AO slices. These slices are integrated into one piece of knowledge that is an abstract context.

Unlike AO, the main purpose of the abstract context is to capture knowledge relevant to the tasks to be solved in an emergency situation rather than to provide a well-formed knowledge classification. Top-level classes of the domain-related knowledge included in the abstract contexts indicate what type of knowledge the classes of the bottom level belong to. Top-level classes specify inherited attributes for the classes of lower taxonomical levels. The classes of the bottom level are the most specified ontology classes. They provide a complete specification for their instances. Classes that in the AO refer to Web-services are included in the abstract context along with these references.

An exemplified piece of domain knowledge included in the abstract context produced for fire as the type of disaster is shown in Figure 3. Classes representing the task knowledge are collapsed in the class *Disaster Relief*. This class includes the task knowledge of the AO (Figure 1). As soon as an abstract context has been produced Web-services, references to which have been included in this context, become aware which tasks to be solved in the current situation. They organise a network of Web-services based on the analysis of interdependencies between the input / output arguments of the tasks that they implement. These interdependencies are revealed from the task hierarchy.

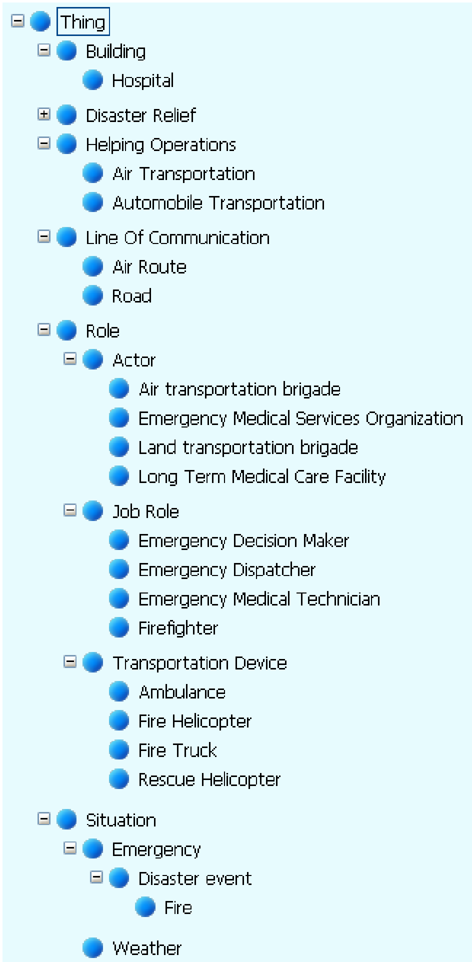

Figure 3. Domain knowledge included in abstract context: taxonomy view

The network of Web-services organised for the purpose of solving the tasks represented in Figure 1, Figure 2 looks as it is shown in Figure 4. In order to make the figure readable some tasks are represented by their super-tasks only. Arrows in the figure depict execution sequences of the Web-services. The tasks implemented by Web-services organising parallel paths can be solved simultaneously. The same names of Web-services shown under different tasks mean that these tasks are implemented as different functions of the same Web-service.

In Figure 4 the Web-services implementing tasks concerned with supplying the DSS to data from information sources use the following kinds of information sources. The current weather conditions (*Get precipitation*, *Get temperature*, *Get wind speed and direction*, and *Get visibility*) are taken from sensors and Web-sites. Information about roads of the region (*Get road*) and their floodability (*Road floodability*) is taken from the GIS. Information about location of the disaster, its type, and approximate number of victims is provided by the user. Information about hospitals available in the region and their locations is read from a database, hospital free capacities are provided by a hospital administration system.

Below, a simplified example of CSP for determination if a road available or not under the current weather conditions is shown. A set of solutions for the road availability task will comprise a set of available roads. CSP in general form is represented as (V, D, C), where V – a set of variables, D – a set of variable domains, C – a set of constraints have to be satisfied.

In the example clause (1) specifies that in the abstract context there is class *Road* with attribute *Available* that can take two values: True or False (the values are represented by d_{11} domain). Constraints c_{11} and c_{12} restrict the domain of the attribute. They specify that only instances of class *Road* that have True in the attribute value are of interest (c_{11}) and that the value for this attribute is determined by *Route availability* task (c_{12}).

$$v_{11} \in V: \texttt{[Road].[available]};\ d_{11} \in D: \texttt{[Road].[available]} \in \{\texttt{True, False}\};$$
$$c_{11} \in C: \texttt{[Road].[available]=True},\ c_{12} \in C: \texttt{[Road].[available]=}\textit{Route availability}\texttt{([Road].[flooded])} \quad (1)$$

$$v_{21} \in V: \texttt{[Road].[flooded]};\ d_{21} \in D: \texttt{[Road].[flooded]} \in \{\texttt{True, False}\};$$
$$c_{21} \in C: \texttt{[Road].[flooded]} = \textit{Road floodability}\ \texttt{([Road].[floodable], [Weather].[precipitation])} \quad (2)$$

$$v_{31} \in V: \texttt{[Road].[floodable]};\ d_{31} \in D: \texttt{[Road].[floodable]} \in \{\texttt{True, False}\} \quad (3)$$

$$v_{41} \in V: \texttt{[Weather].[precipitation]};\ d_{41} \in D: \texttt{[Weather].[precipitation]} \in (0, 100);$$
$$c_{41} \in C: \texttt{[Weather].[precipitation]} = \textit{Get Precipitation}\ () \quad (4)$$

In turn, *Route availability* task has the current state of a road as its input argument. In the constraint c_{12} road state is represented as attribute *Flooded* of class *Road*. Clause (2) specifies attribute *Flooded* of class *Road*, domain of this attribute and a constraint related to the attribute. This constraint says that the value for attribute *Flooded* is determined by task *Road floodability*. Input arguments of this task are values for attribute *Floodable* of class *Road* and for attribute *Precipitation* of class *Weather*. In other words to solve task *Road floodability* it is needed to be aware of if the road is floodable and how much the precipitations are intensive.

Correspondingly, clauses (3) and (4) give specifications for attributes *Floodable* and *Precipitation* and classes these attributes belong to. Additionally, clause (4) specifies that the value for attribute *Precipitation* is determined by task *Get Precipitation*. This task returns information about the precipitation intensity using rating scale (0, 100).

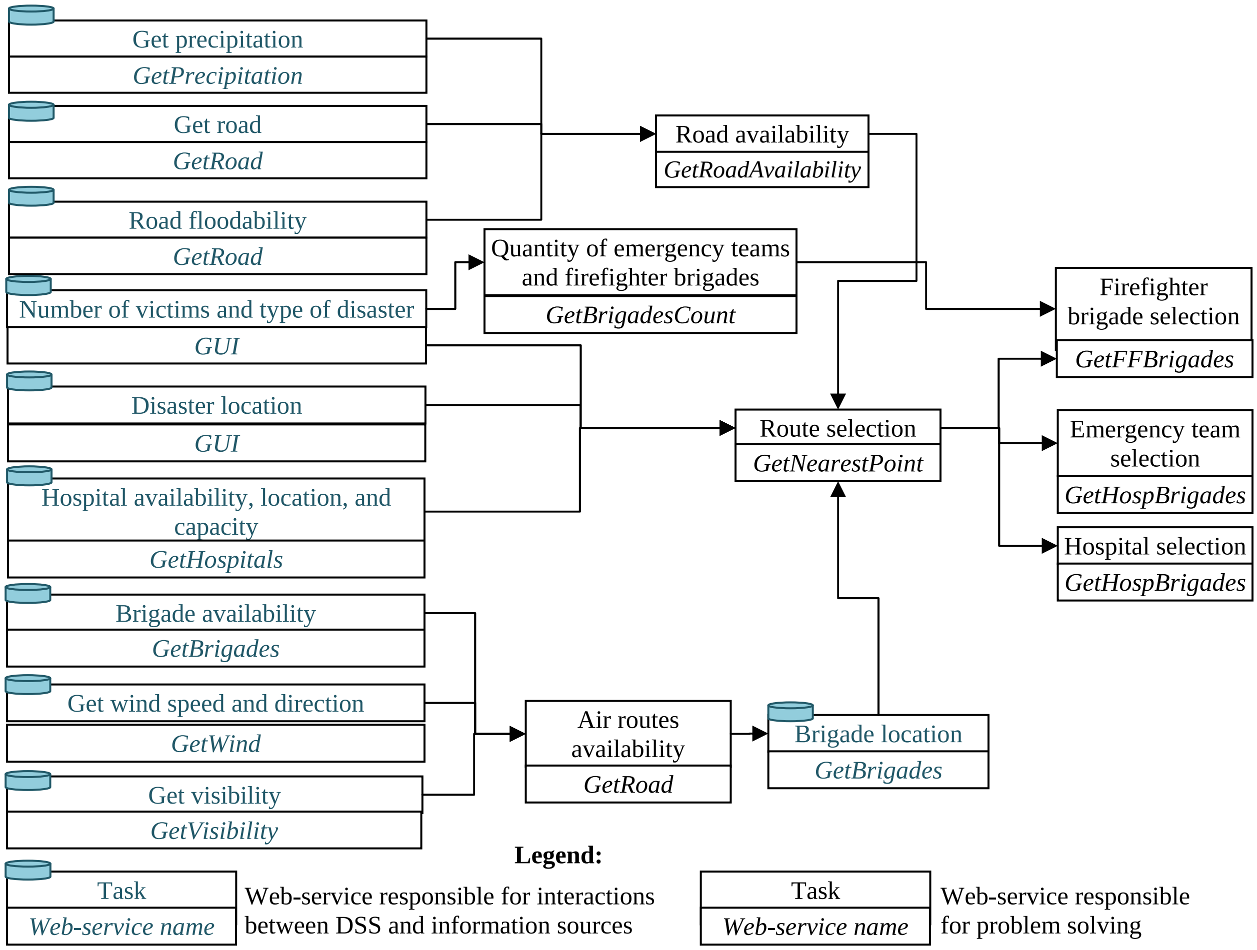

Figure 4. Network of Web-services for fire response

Multi-Role Decision Support

Personalisation of the user support requires maximization of the amount of useful information presented to the user, on the one hand, and minimization of the difficulty the user has in information assessment, on the other hand. The presented research comes to this problem introducing in the AO the class *Job Role*.

Subclasses of the class *Job Roles* are roles filled by persons in an event while performing their job activities. Each job role is assigned a set of tasks pertinent to it. The assignment is specified as associative relationships between the class corresponding to a role and the classes corresponding to tasks this role deals with. As a result, the job roles serve as an additional constraint imposed on the operational context. The user is presented part of the operational context that provides information for tasks for which the user of this particular role is allowed to make decisions. In other words, this part contains information useful (or allowed) for the particular user.

In the abstract context produced for fire response the class *Job Role* is represented by its subclasses *Emergency Decision Maker*, *Emergency Dispatcher*, *Emergency Medical Technician*, and *Firefighter*. The scenario of multi-role decision support under actual conditions of the fire response operation is demonstrated in the next Section.

OPERATIONAL CONTEXT

Operational context integrates information provided by the Web-services that have organised the service network. The Web-services provide to the operational context information taken from the information sources and information calculated by the DSS based on the task constituent.

The user is presented with the operational context after the instantiation of the user role. As soon as a user has authenticated in the DSS the information about the role of this user is taken from his / her profile.

The emergency dispatcher has to be aware of the place of the disaster. His / her activities concern informing appropriate people that a fire takes place and characterising the fire. So, the *emergency dispatcher* is presented with the map of the region. He / she enters a request about the disaster, enters the type of the disaster (fire), points its location on the map, enters potential number of victims and additional descriptions of the event. Based in this request an abstract context is built.

When the *emergency decision maker* logs in into DSS he / she sees a list of emergency situations requiring decisions, characteristics of the situations (data, time, location, etc. characterising the situation), and situation statuses (completed, current, declined decisions). When the *emergency decision maker* activates a situation corresponding to the fire from the list, the corresponding abstract context is filled with data values. Thus an operational context that represents the development of affairs during the fire event is produced. This operational context (Figure 5 without dotted transportation lines) is presented to the *emergency decision maker* with all the information that is known at a particular moment. In Figure 5, information shown close to the place of the fire represents the fire characteristics. They correspond to the values for attributes of class *Fire*. As well, the current locations of the emergency teams (ambulances and rescue helicopters), the locations of the firefighter brigades (fire trucks and helicopters), flooded roads, and the hospitals available for receiving the injured people are shown. In the operational context changeable information as the current locations of the emergency teams and firefighter brigades, roads availability, weather conditions, etc. is displayed as the changes become available.

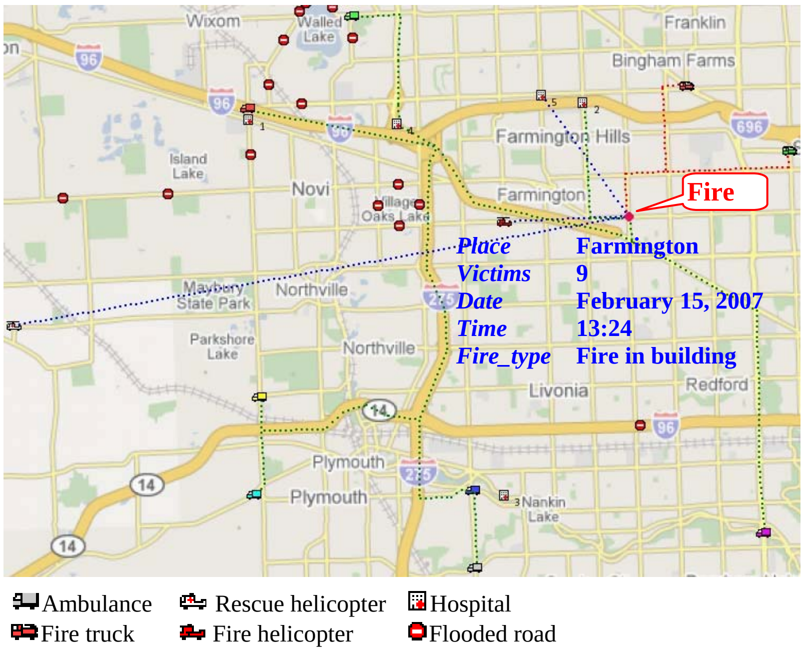

Figure 5. Part of operational context picture with plan for actions

Additionally, the *emergency decision maker* is provided with options to problem solving for generation of action plans for participants involved in the fire response operation. The participants are *Emergency Medical Technician* and *Firefighters.* Solution for the task of designing a plan of actions is a joint solution for the tasks *Firefighter brigade selection*, *Emergency team selection*, *Hospital selection,* and *Route selection*. Before the DSS starts solving this task it provides to the *emergency decision maker* some criteria as minimal cost of the response operation or transportation in the shortest possible time. Then the DSS applying the constraint satisfaction technology generates a set of feasible alternative solutions for the tasks above. The *emergency decision maker* makes the decision choosing one of the solutions from the generated set.

A solution generated by the DSS for the tasks *Firefighter brigade selection*, *Emergency team selection*, *Hospital selection*, and *Route selection* and chosen by the *emergency decision maker* is shown in Figure 5 by dotted lines. These lines show ways of transportations of the selected teams and brigades. The solution is generated for the criterion of the shortest possible time of injured people transportation and arrival to the fire place. This solution is delivered to other participants of the operation. According to the multi-role decision support, the participants can either accept or decline the plan. For example, if a leader of one of groups of the participants does not have an opportunity to put the rescue helicopter at the disposal then he /she declines this decision. In this case DSS removes the rescue helicopter from the operational context and returns the renewed operational context to the *emergency decision maker* so that he / she could regenerate the set of solutions.

CONCLUSION

The research presented in the paper is primary oriented on context-aware operational decision support. The research results show that the proposed hybrid technology makes it possible to obtain ontology-based problem model that can be interpreted as constraint satisfaction problem and to provide multiple users with information corresponding to their interests, goals, and functions in decision making. The ontology-based problem model represented as context provides the Web-services with awareness about the problems to be solved in the current situation and information needed for this. Ontology-based referencing of Web-services proposed in the research allows the Web-services to exchange information about their needs and possibilities in terms on the ontology vocabulary and organise themselves in a network of Web-services for a common purpose.

ACKNOWLEDGMENTS

This research is supported in part by CRDF partner project RUM2-1554-ST-05 with US ONR and US AFRL, projects supported by the Russian Academy of Sciences 14.2.35 and 1.9, and grants 06-07-89242 and 08-07-00264 of the Russian Foundation for Basic Research.

REFERENCES

1. Chapman, S. and Ciravegna, F. (2006) Focused Data Mining for Decision Support in Emergency Response Scenarios, *Proceedings of ISWC-2006 Workshop on Web Content Mining with Human Language Technology*, Athens, USA, available at http://orestes.ii.uam.es/workshop/18.pdf.
2. Gouman, R., Kempen, M., de Vree, P., Capello, T., van der Heijden, E. and Wijngaards, N. (2007) The Borsele Files: The Challenge of Acquiring Usable Data under Chaotic Circumstances, *Proceedings of the 4th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2007),* Delft, the Netherlands, 93 – 103.
3. Harrald, J. R., Jefferson, T. I., Fiedrich, F. and Sener, S. (2007) A First Step in Decision Support Tools for Humanitarian Assistance during Catastrophic Disasters: Modeling Hazard Generated Needs, *Proceedings of the 4th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2007),* Delft, the Netherlands, 51 – 56.
4. Little, E. G. and Rogova, G. L. (2006) An Ontological Analysis of Threat and Vulnerability, *Proceedings of the 9th International Conference on Information Fusion*, Florence, Italy, paper no. 302.
5. Löffler, J., Hernández, V. E., Schon, J., Pottebaum, J. and Koch, R. (2007) Intelligent Use of Geospatial Information for Emergency Operation Management, *Proceedings of the 4th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2007)*, Delft, the Netherlands, 181 – 190.
6. Potter, S., Kalfoglou, Y., Alani, H., Bachler, M., Buckingham Shum, S., Carvalho, R., Chakravarthy, A., Chalmers, S., Chapman, S., Hu, B., Preece, A., Shadbolt, N., Tate, A. and Tuffield, M. (2007) The Application of Advanced Knowledge Technologies for Emergency Response, *Proceedings of the 4th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2007)*, Delft, the Netherlands, 361 – 368.
7. Smirnov, A., Pashkin, M., Chilov, N. and Levashova, T. (2003) KSNet-Approach to Knowledge Fusion from Distributed Sources, Computing and Informatics, 22, 105 – 142.
8. Smirnov, A., Pashkin, M., Chilov, N., Levashova, T. and Krizhanovsky, A. (2003) Ontology-Driven Knowledge Logistics Approach as Constraint Satisfaction Problem, *Proceedings of the International Conference on Ontologies, Databases and Applications of Semantics Information Reuse and Integration (ODBASE'2003),* Catania, Sicily, Italy, LNCS, 2888, 635 – 652, Springer, Berlin, Heidelberg.
9. Smirnov, A., Pashkin, M., Shilov, N. and Levashova T. (2007) Intelligent Support of Context-Based Megadisaster Management: Hybrid Technology and Case Study, *Proceedings of the 4th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2007)*, Delft, the Netherlands, 305 – 316.
10. Van Someren, M., Netten, N., Evers, V., Cramer, H., Hoog, R. and Bruinsma, G. (2005) A Trainable Information Distribution System to Support Crisis Management. *Proceedings of the 2nd International Conference on Information Systems for Crisis Response and Management (ISCRAM 2007)*, Brussels, Belgium, 203 – 206.
11. White, C., Turoff, M. and Bartel Van de Walle (2007) A Dynamic Delphi Process Utilizing a Modified Thurstone Scaling Method: Collaborative Judgement in Emergency Response, *Proceedings of the 4th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2007)*, Delft, the Netherlands, 7 – 15.