Intelligent Support of Context-Based Megadisaster Management: Hybrid Technology and Case Study

Alexander Smirnov, Mikhail Pashkin, Nikolai Shilov, Tatiana Levashova
St.Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences,
St.Petersburg, Russia
{smir, michael, nick, oleg }@iias.spb.su

ABSTRACT

The situation with the hurricane Katrina showed that the conventional tiered response to disaster event, whereby state and local officials are responsible for the first few days, does not work well in case of megadisasters (massive hurricanes, earthquakes, large-scale acts of terrorism, etc.). Such situations require application of new technologies for preparing the operation, interoperability between the operation participants, and decision support for officials. Here presented approach proposes a context-driven decision support schema based on integration of such technologies as context & ontology management and constraint satisfaction. The application of the approach is illustrated via a case study of a portable hospital arrangement.

Keywords

Megadisaster management, humanitarian logistics, intelligent decision making support, context management.

INTRODUCTION

The number of annual natural and human-made disasters (including terrorism) has tripled since 1970. The strains on humanitarian organizations responding to emergencies showed that 256 million people were reported affected by disasters in 2004, while the annual average is 210 million (Fritz Institute, 2005). Furthermore, in recent years we experienced several "megadisasters" such as hurricane Katrine and tsunami in Indian Ocean in 2004 (earthquakes and large scale acts of terrorism can also be related to such kind of disasters).

Dramatic effects of megadisasters showed that the conventional tiered response to disasters does not work well in this case. Fast (time-critical) response to megadisaster events (massive hurricanes, earthquakes, etc.) requires emergency preparedness based on long-term response scenarios planning with realistic (or predictable) expectations concerning available (alternative) federal & local sources and estimation of access time to them. Megadisasters require application of new technologies including context-aware interoperability of participants, on-the-fly decision support assistance and other. The practice shows that one of the most difficult steps in responding for such situations is providing for the right relief supplies to the people in need at the right time. At the same time delivering of too much supplies or wrong supplies means loosing time and money. Therefore, a new research direction “humanitarian logistics” standing for *processes and systems involved in mobilizing people, resources, skills and knowledge to help vulnerable people affected by natural disasters and complex emergencies*, has appeared as a key enabler for successful disaster relief (Fritz Institute, 2005). This fact motivated the choice of the case study for implementation of the presented here approach.

In the paper the problem of portable hospital arrangement is considered as a case study. This problem is very complex and includes tasks from such areas as logistics, healthcare and other. Solving these tasks will require intensive usage of knowledge. Importance of intelligent systems in the area of operations management has been widely recognized recently, particularly, in the context of decision making.

The following requirements to decision support systems for such problem domains have been identified:

- robustness: the system should continue to operate even if some of its elements stop;
- sensitivity and adaptability: relationships between the system's elements (organization's units) have to be able to be easily and quickly readjusted in accordance with changes in the environment;

- intensive knowledge / information exchange between the system's elements: knowledge management de facto has become essential for decision making processes.

Among the various types of system architectures the networked distributed architecture best meets the above requirements (Scerri et al., 2005). Such systems link together independent people, assets and ideas to work together for a common purpose; they have multiple leaders, lots of voluntary links and interacting levels.

Proposed here approach is based on the developed earlier concept of knowledge logistics (Smirnov et al., 2004) that stands for acquisition of the right knowledge from distributed sources located in a network-centric environment, and integration and transfer of this knowledge to the right person within the right context, at the right time, for the right purpose. The approach applies knowledge logistics to support decision making processes. It is based on context management, knowledge acquisition from available knowledge sources, and current situation analysis.

The rest of the paper is organized as follows. The methodology proposed is described in the next section. It is based on usage of ontological knowledge and contexts of two types: abstract and operational. Next a hybrid technology framework is described. It is tested through a case study experiments presented then. Some findings and results are summarized in the conclusion and future work is proposed.

PROPOSED METHODOLOGY

The presented methodology is intended for operational decision support. It proposes integration of environmental information and domain knowledge in a context of current situation through linkage of representation of this knowledge with semantic models for environmental information sources providing information about the environment.

Ontologies are widely used for problem domain description in the modern information systems. Ontology is an explicit specification of a structure of a certain domain. It includes a vocabulary for referring to notions of subject area, and a set of logical statements expressing the constraints existing in the domain and restricting the interpretation of the vocabulary (FIPA, 2007).

The methodology (Figure 1) considers context as a problem model based on the knowledge extracted from the problem domain and formalized within an application ontology by a set of constraints. The set of constraints, additionally to the constraints describing domain knowledge, includes information about the environment and various preferences of the user concerning the problem solving (user defined constraints). The methodology takes into account different user roles as different levels of user responsibility. The problem is suggested being modeled by two types of contexts: abstract and operational. Abstract context is an ontology-based model integrating information and knowledge relevant to the problem. Operational context is an instantiation of the abstract context with data provided by the information sources.

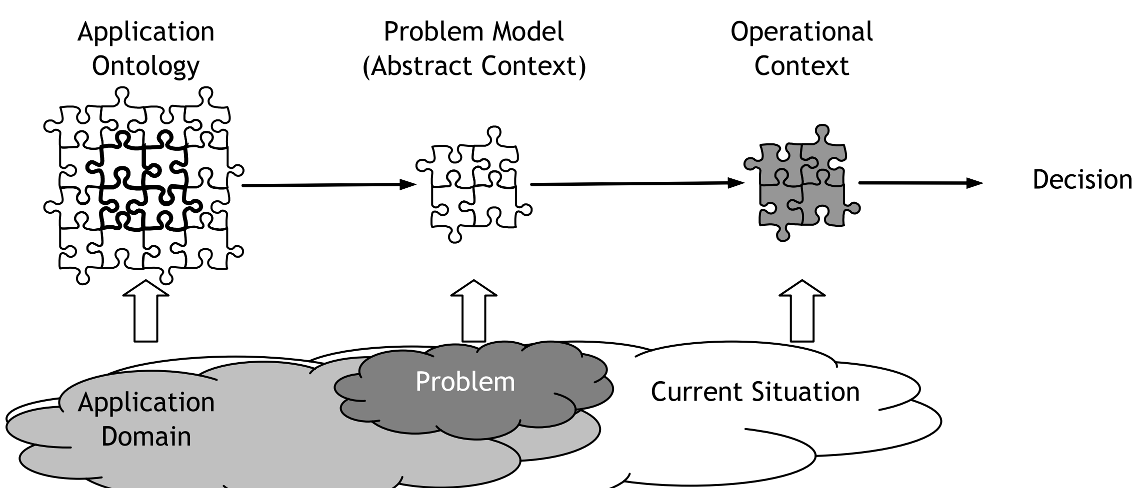


Figure 1. Context-based decision support

The presented methodology is aimed at building a context model representing the current situation or problem to be solved by the decision maker. Within the paper current situation and problem will be referred to as problem; the decision maker will be referred to as the user. Context describes factors influencing the current situation and provides requirements to solutions to be generated for the decision maker (e.g., requirements to coordination of multinational, multicultural efforts in countries with little or no infrastructure). Figure 2 presents an example of fragments of both types of contexts used for problem modeling: *abstract context* on the left side and *operational context* on the right side. It can be seen that attributes “x-coordinate”, “y-coordinate” and “cost” are assigned values 246, 310 and 1000 respectively.

The problem is suggested being modeled by two types of contexts: abstract and operational. *Abstract context* (Figure 2, left) is an ontology-based model integrating information and knowledge relevant to the problem. *Operational context* (Figure 2, right) is an instantiation of the abstract context with data provided by the information sources. In Figure 2 it can be seen that attributes “x-coordinate”, “y-coordinate” and “cost” are assigned values 246, 310 and 1000 respectively.

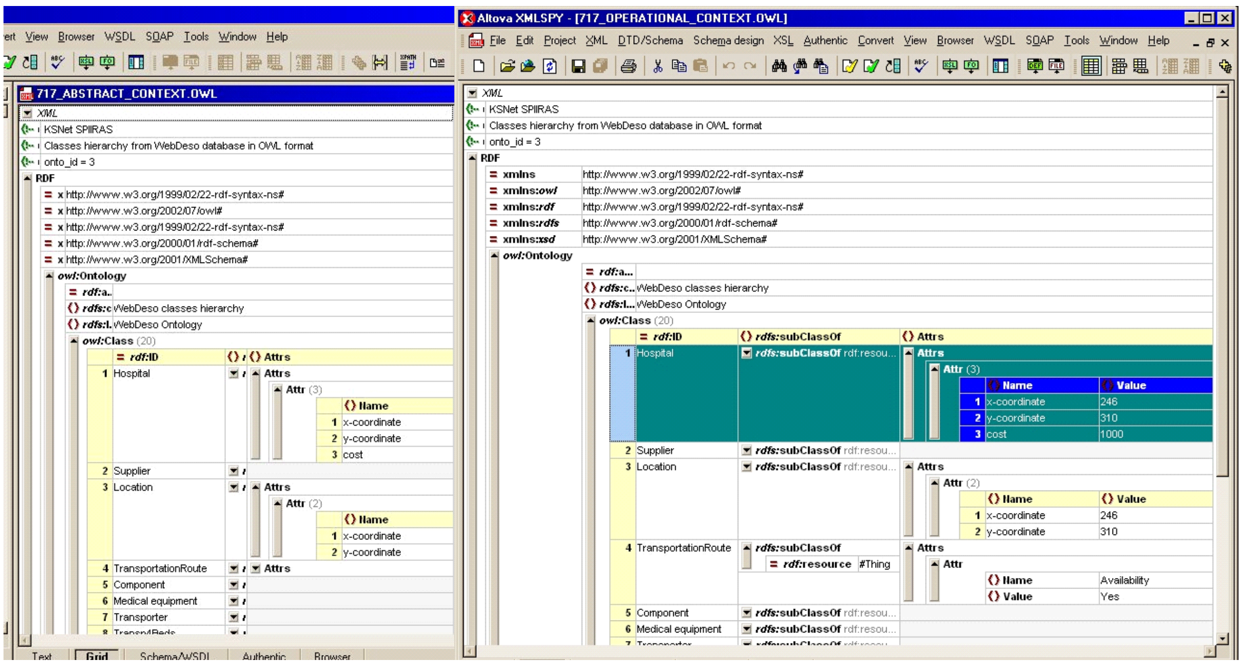


Figure 2. Example of XML representation of abstract context (left), and operational context (right)

The problem is supposed to be interpreted as dynamic constraint satisfaction problem (CSP) by a constraint solver. CSP model consists of three parts: (i) a set of variables; (ii) a set of possible values for each variable (its domain); and (iii) a set of constraints restricting the values that the variables can simultaneously take. Solution of CSP is a set of values of variables satisfying constraints of variables (Tsang, 1993, Mackworth, 1992).

In order to provide compatibility of the ontology model for knowledge representation and internal solver representations a formalism of object-oriented constraint networks (OOCN) is used (Smirnov et al., 2003b) for ontology representation. Compatibility of CSP, ontology, and OOCN models is achieved through identification of correspondences between primitives of these models. Typical ontology modeling primitives are classes, relations, functions, and axioms. The formalism of OOCN describes knowledge by sets of objects, variables, possible values for each variable, and constraints restricting the values of the variables and describing relations between objects. The concept “class” in the proposed notation is used instead of the concept “object” and the concept “attribute” is introduced instead of the concept “variable” in the way object-oriented languages suggest.

As a result, ontology-based problem model is described by a set of constraints and can be directly mapped into the constraint solver. A result of CSP solving is one or more satisfactory solutions for the problem modeled. Correspondences between the primitives of ontology model, OOCN, and CSP is shown in Table 1.

Ontology Model	OOCN	CSP
Class	Object	Set of variables
Attribute	Variable	
Attribute domain (range)	Domain	Domain
Axioms and relations	Constraints	Constraints

Table 1. Correspondence between ontology model, OOCN, and CSP model

Decision making deals with complex problems expecting deep knowledge of the domain. The users do not necessarily have satisfactory knowledge. This fact is the most important at the operational level when the user has to make decisions under time pressure. Because of this, the approach relies on an availability of sufficient domain knowledge and support of subject experts, if required. The domain knowledge is collected before it can be used in decision making.

Domain knowledge is modeled by ontologies of three types: domain ontology, tasks & methods ontology, and application ontology. Domain ontology represents conceptual knowledge about the domain, tasks & methods ontology formalizes tasks identified for the domain and hierarchies of problem solving methods (taking into account alternative ones). An ontology consists of classes, class attributes, attribute domains, and constraints. The constraints represent "class-attribute-domain" relation, structural relationships (hierarchical, "part-of" and "is-a"), classes compatibility, associative relationships, class cardinality restrictions and functional dependencies. The tasks and methods are represented by classes; the sets of methods' arguments and argument's types are represented by sets of attributes and domains, respectively. Domain and tasks & methods ontologies are interrelated by relationships that specify values of which class attributes of the domain ontology serve as input arguments for the methods of the task & methods ontology. Application ontology is a specialization of domain and tasks & methods ontologies. Knowledge from domain and tasks & methods ontologies is integrated into application ontology that describes a real-world problem domain depending on particular domain and problem. Ontologies of these types are stored in the ontology library.

HYBRID TECHNOLOGY FRAMEWORK

The technology framework incorporates technologies of *profiling*, *ontology management*, *context management*, and *constraint satisfaction* (Figure 3). Profiling is used for the personalized user support. Ontology management techniques are used to perform the extraction of relevant knowledge, its integration and consistency checking. In order to enable capturing, monitoring, and analysis of decisions and their effects the contexts representing problem models and respective decisions made are retained. For this, context versioning techniques are applied. As a result the user is provided with reusable problem models and knowledge of similar situations and decisions made. Obtaining information, its organization in contexts, and context versioning are context management issues. OOCN formalism serves for problem definition, and constraint satisfaction technology is used for problem solving.

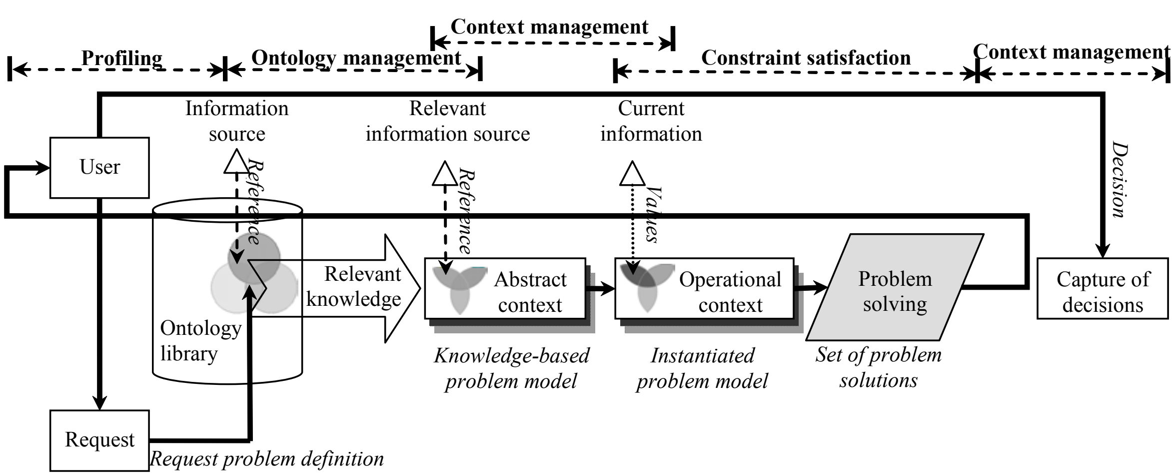


Figure 3. Hybrid technology framework for context-based operational decision support

The operational decision support in the presented methodology relies on a preliminary stage where the presented below components of decision support system (DSS) are built.

DSS implementing the methodology framework is proposed to have distributed architecture (Figure 4). DSS is implemented as a Web-service for the users (decision makers). Main components are scenario rules, ontology library, knowledge map, user profile, and a set of Web-services.

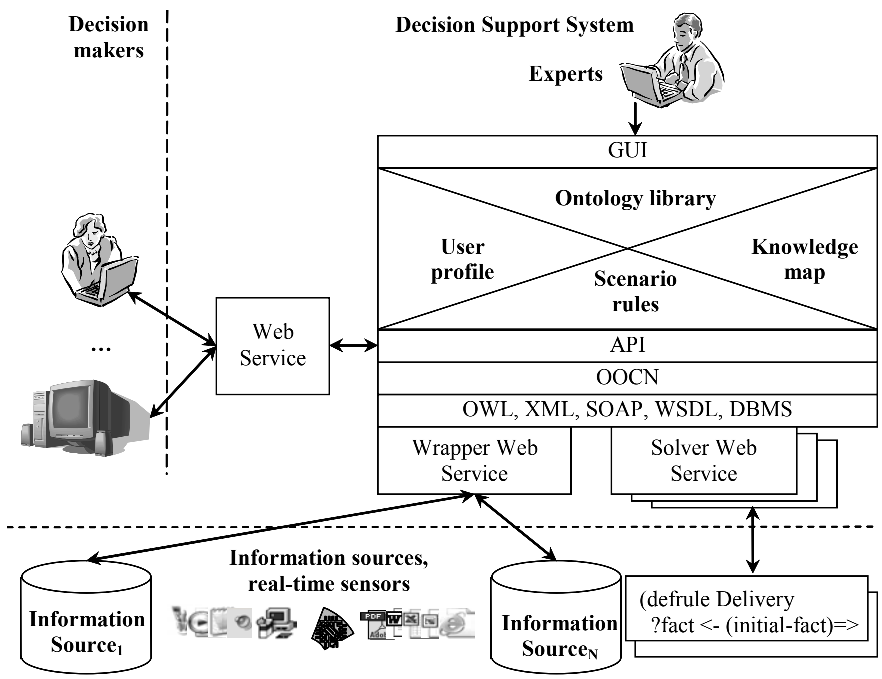


Figure 4. DSS architecture

Ontology library is an internal knowledge storage. It stores ontologies imported from distributed heterogeneous sources (e.g., OWL / DAML+OIL / RDF ontologies available in the Internet, various classifiers, rule and knowledge bases) and built by ontology engineers. Software tools for importing ontologies from the above formats have been developed. The ontologies are formalized in a uniform way. A result of the import is that the imported ontologies are described by means of the internal ontology formalism and the vocabulary supported by the ontology library (Smirnov et al., 2003a). References to the information sources used for instantiation of ontology classes are organized in a knowledge map. It contains such properties of information sources as metadata, accessibility, location, format and other.

Semantics of information sources is described via information source capabilities model; domain knowledge is modeled via ontology model; users are modeled via user profile model. All the components are represented by the OOCN formalism that is considered as the internal knowledge representation. The common representation of the components allows unification of information and knowledge provided by the components, enables DSS to handle the information sources in a same manner, and simplifies integration of the information and knowledge.

In order to obtain up-to-date information from the environment, ontologies are linked to information sources that keep track of environment changes. Domain knowledge, information sources and users are coupled through the attributes. For this, attributes of domain ontology and attributes of the representations for information sources and users are linked by associative relationships. The links mean that the attribute of the ontology class gets values provided by the information source or user.

Scenario rules are responsible for calling the system scenarios. The scenarios are implemented as a sequence of functions called one-by-one inside the main program module. For instance, to solve a problem of mobile hospital arrangement a set of subtasks has to be solved: (i) hospital allocation, (ii) finding of suppliers, and (iii) transportation task. Results obtained during solving one task can be initial data for another task and they have to be stored. Solutions of the problem are generated by the system and presented to the user. When the user makes the final decision the system performs a set of actions: generates messages to operation participants, updates profile of the decision maker, etc. Scenario rules describe sequences of these tasks and relations between them.

Based on the assumption that one constraint solver is not enough to solve complex problems occurring in real-world domains the complex problems are proposed to be solved coordinating several constraint solvers. The solvers installed at different computers can be used while problem solving. This is the case of distributed problem solving. In the proposed architecture the constraint solvers are implemented as Solver Web Services.

Using graphical user interfaces, the subject experts add ontologies into the ontology library and modify them if required; prepare representations for the information sources; align the ontologies and the information source representations; modify and validate scenario rules; and analyze information accumulated in the user profiles to define users' preferences that cannot be determined automatically.

Wrapper Web Services are responsible for the interaction with information sources in the following ways:

- representation of information provided by information sources by means of the OOCN-formalism;
- querying information sources;
- transfer of the information to the DSS;
- information integration;
- data conversion.

The following types of information sources are distinguished:

- sensors are physical devices that detect a signal, physical conditions, etc; Web-services working with sensors have to support their outputs and interfaces.
- databases are organized collections of information; Web-services that work with databases have to support SQL queries;
- Web-sites and RSS (RDF Site Summary) are textual information with predefined structure; Web-services that work with this sources have be able to work with RSS structure;
- users may know large amount of information and can pass it to the Web-service through graphical user interface (GUI);
- other information sources allowing interaction through Web-services or for which appropriate Web-services can be developed.

Users work via GUI connected to a Web-service that processes the user request and sends it to the DSS. Information about the user required to describe the situation represented in the abstract and operational contexts can be taken from the user profile or directly from the user. In the former case the user is considered as an information source supporting pull technology. In the latter case the user is thought of as supporting push technology.

The presented technological framework has been implemented as a research software prototype. The prototype covers tasks from the following areas: (i) ontology management: import, modification, storage, and presentation; (ii) context management: building of contexts of both types, storage, validation, and versioning of context; (iii) profiling: building, update, and analysis; (iv) tasks automation: input of requests, processing of information sources, and solving CSP using constraint solvers. Integrated scenario executed using the developed framework is presented in the next section.

CASE STUDY

A portable hospital can be characterized as a portable structure, which is supposed to be built within a limited time with given characteristics such as capacity and location of the hospital, construction expenditure, etc. A general request to DSS has the following format: "*Define suppliers, transportation routes and schedules for building a portable field hospital of a given capacity at a given location by a given time*". The scenario considered comprises two major stages (Smirnov et al., 2006). The preliminary stage enables preparedness to an emergency situation. At this stage information sources to be used are selected and linked. These sources have to provide for all required information needs including systematic monitoring and evaluation of transport infrastructure, analysis of historical meteorological records of the region, monitoring of new construction works and changes in existing infrastructure, search for possible suppliers, etc. All these activities have to continue at the decision making stage as well. At the decision making stage operational decisions are made. Links to the information sources allow the system to deal with real-time information such as state of the roads, waterways and other transport infrastructure, availability of supply storage facilities and means of transport, suppliers' inventory and stock levels, etc. Conditions that different supplies have different lifetime, storage and transportation conditions are taken into account, as well. For example, tents and clothing have to be stored in dry places but do not have strong restriction on expiration date, whereas medicines may have short expiration time and strict storage requirements.

Preliminary Stage

Parts of ontologies corresponding to the task formulated in the request were found in Internet ontology libraries (Cyc, 2007, Loom, 2007, WebOnto, 2007). These ontologies represent a hospital in different manners using different representation formats. Firstly, the ontologies were imported from the source formats into the system notation. Next, ontology parts relevant to the request were combined into a single ontology. Then ontology engineers slightly modified this ontology producing the common application ontology (Figure 5). The figure legend is as follows: shadowed boxes contain classes imported from tasks & methods ontology, firm unidirectional arrows represent hierarchical “is-a” relationships, dotted unidirectional arrows represent hierarchical “part-of” relationships, double-headed arrows show associative relationships.

As a result of the analysis of the problem formulated in the request a number of interrelated modules (subproblems) were defined (Figure 6). The figure legend is as follows: bold label denotes the common problem, underlined labels denote subproblems and italic labels denote example parameters common for two or more subproblems. Based on these subproblems abstract contexts are identified.

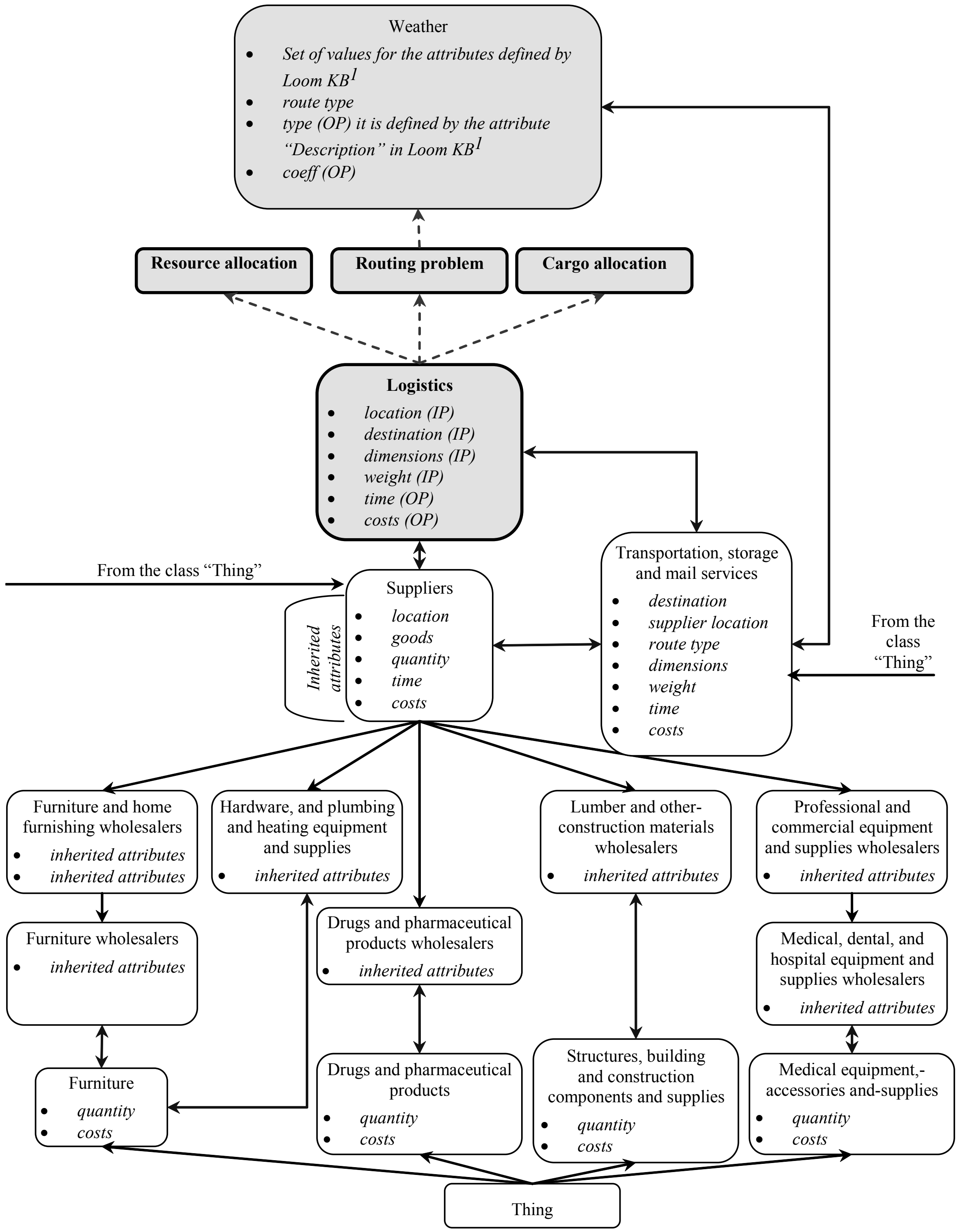
To provide a decision maker with required information the system has to execute a certain scenario. In the presented case study the scenario is described by sequence of exemplified tasks to be solved within the tasks & methods ontology (Table 2): (i) define the number of emergency teams and firefighter brigades taking into account type of disaster and quantity of victims; (ii) determine route availabilities based on the current weather conditions; (iii) select emergency teams for taking injured people to hospitals based on the current team locations, team availabilities, hospital locations and availabilities, and route availabilities; and (iv) select firefighter brigades for extinguishing the fire and means for their transportations based on the current brigade locations and availabilities, and route availabilities. Tasks of the same row are implemented within one Web-service. During the scenario execution the abstract and the operational contexts are built.

Simplified examples of the identified abstract contexts based on the domain ontology for the subproblems outlined with dashed lines in Figure 6 are presented in Figure 7. Part (a) illustrates abstract context for "Resource Allocation" subproblem, part (b) illustrates abstract context for "Hospital Allocation" subproblem, and part (c) illustrates abstract context for "Routing" subproblem. Rectangles denote classes with attributes, solid lines denote associative relationships. Grayed classes and relationships are not included into the contexts. Unlike slices, abstract contexts include references (not shown in Figure 7) to information sources with actualized information.

Decision Making Stage

Figure 8 illustrates operational context built on the basis of the abstract context for the "Resource Allocation" subproblem. Unlike the abstract context, it does not contain classes but instances of these classes generated on the basis of the information from information sources (databases, sensors, experts, etc.). Values with question marks are to be calculated based on the functional constraints or (in case of the hospital costs) are the objective functions for evaluation of solution feasibility.

Due to OOCN knowledge representation formalism has been chosen, the operational context is a formalized CSP. Since the problems to be solved are presented in the same formalism but may differ in structure and in values of variables, an on-the-fly problem modification and solving based on adaptive software modules are proposed. For this purpose the described approach implements "adaptive services" that can modify themselves when solving a particular problem described by the operational context. Upon receiving the operational context the service generates an executable module for solving required tasks “on-the-fly”. ILOG constraint satisfaction technology (ILOG, 2007) was chosen as a constraint solver for the implementation of the service, which has been extended by introduction of the contexts and implementation of the approach.



IP – Input Parameters

OP – Output Results

[1] Theory "Weather" in Loom and PowerLoom knowledge bases, Information sciences Institute, The University of Southern California, http://sevak.isi.edu:4676/loom/shuttle.html

Figure 5. Application ontology for portable hospital arrangement (a fragment)

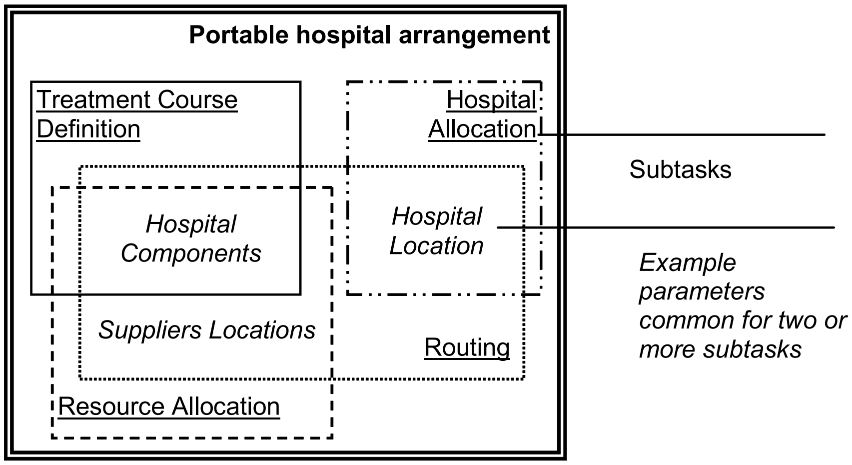


Figure 6. Subproblems of the portable hospital arrangement problem (a simplified example)

Module	Task	Description	Input	Output
GIS	Get point Get Latitude Get Longitude	Returns array of points of the region with their coordinates	Service_URI	Id's and coordinates of points
	Get road Road floodability Beginning of road Ending of road	Returns array of roads of the region with their properties	Service_URI	Id's and "floodability" of roads, connected points (beginning and ending)
Weather	Get temperature	Returns current temperature	Service_URI	Temperature
	Get wind	Returns current wind conditions	Service_URI	Wind speed and direction
	Get precipitations Get visibility	Returns current precipitations and visibility conditions	Service_URI	Precipitations, visibility
Brigades	Brigade availability	Returns list of brigades of the region and service URI's for acquiring their properties	Service_URI	Id's, types and service URI's of available brigades
Brigade	Brigade location Get brigade	Returns current location of a given brigade	Service_URI Brigade id	Location point id
Hospitals	Get hospital availability Get hospital location	Returns a list of hospitals of the region and related information	Service_URI	IDs of the hospitals, their addresses, free capacity, availability
Calculations	Route availability	Checks if a road is currently available or not for a given vehicle type	Road, its id and properties (e.g., flooadbility), vehicle type, weather conditions	Available / not available
	Quantity of emergency teams and firefighters	Calculates required quantity of emergency teams and firefighters	Disaster type Number of victims	Required number of brigades

Table 2. Tasks solved during the system scenario execution

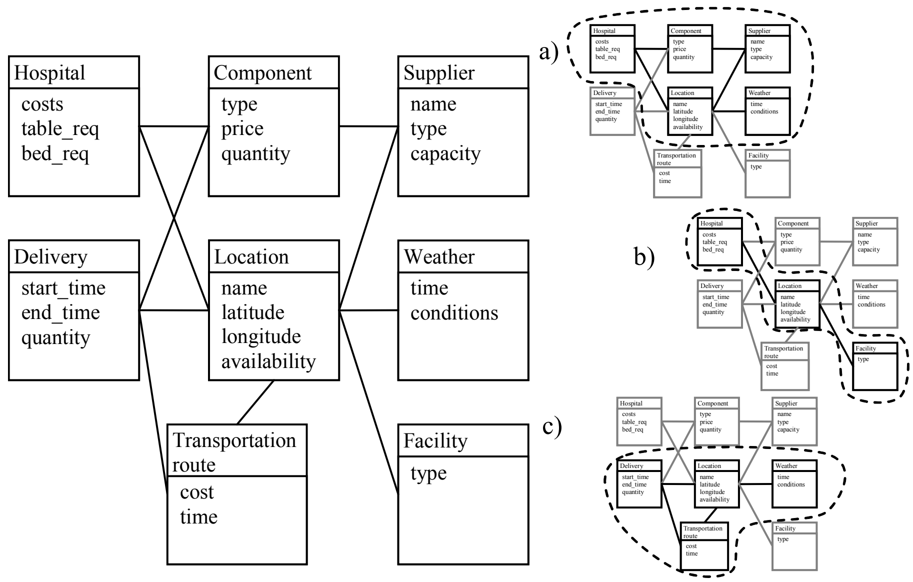


Figure 7. Examples of abstract contexts

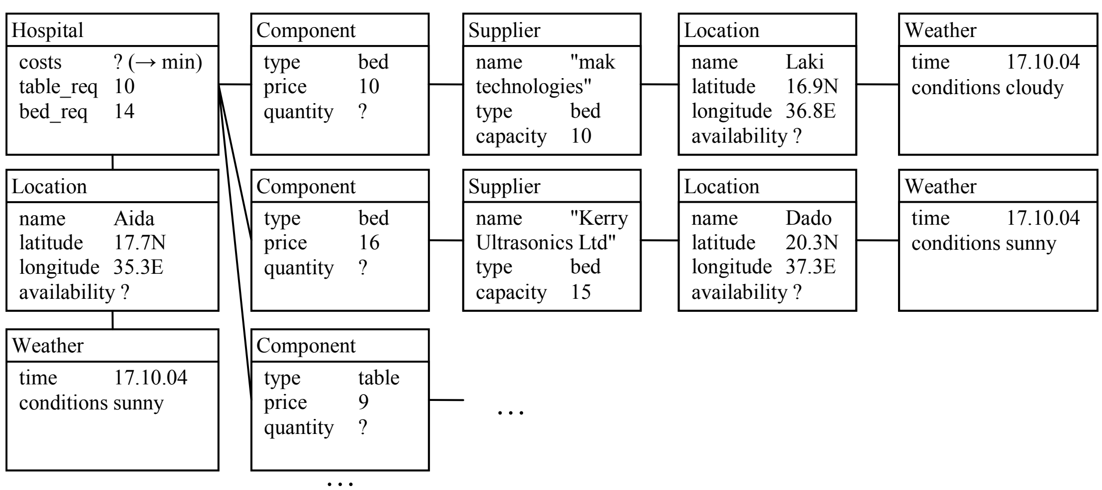


Figure 8. An example of operational context for the "Resource allocation" subproblem

CONCLUSION AND FUTURE WORK

The paper presents an approach to decision making support for megadisaster management. The technological framework of the approach is based on application of such technologies as ontology and context management and constraint satisfaction. Application of ontologies makes it possible to operate with knowledge taking into account semantics; context management technology is used for taking into account the condition of the environment ("problem context") and constraints allow easy formalization of the problem since problems from the areas of planning, scheduling and management are often described as constraint satisfaction problems. The application of the approach is illustrated via a case study of a portable hospital arrangement.

The presented approach has a number of potential advantages for the operational decision making: (1) contexts contain information relevant to a particular task or situation, that allows selecting source types responsible for observation constraints relevant to application area; (2) ontologies make it possible to transform information provided by sources into knowledge at the level of description of application area, therefore an ontology-driven context at the decision making level provides the decision maker with the knowledge; (3) context management technique enables generation of alternative contexts representing alternative situations or alternative ways of problem solving; (4) knowledge representation via object-oriented constraint networks allows working with the operational context as if it was a constraint satisfaction problem and generating feasible solutions using a constraint solver.

Usually megadisaster relief and evacuation tasks involve a large number of different heterogeneous teams (sometimes multinational), which have to collaborate in order to succeed (Smirnov et al., 2005). Such teams might include medical brigades, firefighters, rescuers, military personnel, commercial / governmental / non-commercial organizations, volunteers, etc. Besides, during a rescuing operation it might be necessary to use external sources to get required information (e.g., medical databases, transport availability, weather forecasts). Such organization (networked organization) requires intensive information exchange in order to achieve necessary level of the situational awareness, create ad-hoc action plans, and have continuously updated information.

Centralized control is not always possible due to damages in local infrastructure, different subordination of participating teams, etc. Another disadvantage of the centralized control is its possible failure that would cause stopping of the entire operation. Possible solution for this is organization of a decentralized self-organizing network consisting of the operation participants. However, in order for this network to operate it is necessary to solve a number of problems that can be divided into technical (hardware-related) and methodological constituents. The approach presented is aimed at solving the latter. It will include such subproblems as: providing for semantic interoperability between the participants, definition of the standards and protocols to be used, development of negotiation rules and protocols for ad-hoc decision making and design of a case study for estimating the applicability of the approach.

ACKNOWLEDGMENTS

The paper is due to research carried out as part of CRDF partner project #RUM2-1554-ST-05 with US ONR and US AFRL, projects supported by the Russian Academy of Sciences # 16.2.35 and # 1.9, grants #05-01-00151 and # 06-07-89242 of the Russian Foundation for Basic Research.

REFERENCES

1. Fritz Institute (2005) Humanitarian Logistics: Getting the Right Relief to the Right People at the Right Time, Fact Sheets, *URL*: http://www.fritzinstitute.org/fact_sheets/f_s-hls.html.
2. Scerri, P., Giampapa, J. and Sycara, K. (2005) Techniques and Directions for Building Very Large Agent Teams, *in Thompson, C., Hexmoor, H. (Eds.), KIMAS'05: Modeling, Exploration, and Engineering (proceedings of the 2005 International Conference on Integration of Knowledge Intensive Multi-Agent Systems)*, 79-84.
3. Smirnov, A., Pashkin, M., Chilov, N. and Levashova, T (2003) KSNet-Approach to Knowledge Fusion from Distributed Sources, *Computing and Informatics*, 2003; 22: 105-142.
4. Smirnov, A., Pashkin, M., Chilov, N. and Levashova, T. (2004) Knowledge Logistics in Information Grid Environment, *in Zhuge, H. (Ed.), the special issue "Semantic Grid and Knowledge Grid: The Next-Generation Web" of International Journal on Future Generation Computer Systems*, 20, 1, 61-79.
5. Smirnov, A., Pashkin, M., Chilov, N., Levashova, T. and Krizhanovsky, A. (2005) Agent-Based Intelligent Support to Coalition Operations: A Case Study of Health Service Logistics Support. *Information & Security. An International Journal. Special Issue on IT in Coalition and Emergency Operations*, 16, 41-61.
6. Smirnov, A., Levashova, T. and Shilov, N. (2006) Context-Driven Decision Support for Megadisaster Relief. *Journal of Emergency Management*, 4-5, 51-56.
7. Foundation for Intelligent Physical Agents (FIPA) Documentation (2007). *URL*: http://www.fipa.org.
8. ILOG corporate Web site (2007). *URL*: www.ilog.com.
9. Smirnov, A., Pashkin, M., Chilov, N., Levashova, T., and Krizhanovsky, A. (2003): Ontology-Driven Knowledge Logistics Approach as Constraint Satisfaction Problem. *In: Meersman, R., Tari, Z., Schmidt, D.C. et al. (eds.): On the Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*. *LNCS 2888. Springer Verlag, Berlin Heidelberg New York*, 2003, 535-652.
10. Tsang, E.P.K. (1993) Foundations of Constraint Satisfaction. *Academic Press, London and San Diego, 1993, ISBN 0-12-701610-4*.

11. Mackworth, A.K. (1992) The logic of constraint satisfaction, Artificial Intelligence, Vol.58, Nos.1-3 (Special Volume on Constraint Based Reasoning), 1992, 3-20
12. Hpkb-Upper-Level-Kernel-Latest: Upper Cyc / HPKB IKB Ontology with links to SENSUS. Ontolingua Ontology Server (2007). *URL:* http://www-ksl-svc.stanford.edu:5915.
13. Weather Theory. Loom ontology browser, Information sciences Institute, The University of Southern California (2007). *URL:* http://www.isi.edu/isd/LOOM/.
14. WebOnto: Knowledge Media Institute (KMI), The Open University, UK, (2007). *URL:* http://eldora.open.ac.uk:3000/webonto.