Context-Aware Decision Support in Dynamic Environments: Theoretical and Technological Foundations

Alexander Smirnov, Tatiana Levashova, Nikolay Shilov and Alexey Kashevnik

Abstract The paper addresses the issue of context-aware operational decision support in dynamic environments. The context model specifies conceptual knowledge describing the situation and problems to be solved in this situation. This model comprises knowledge captured from an application ontology, which is formalized by a set of constraints. The context aware decision support system (DSS) developed within the research has a service-oriented architecture. The Web-services constituting the architecture provide the DSS with the contextualized information from information resources, solve problems specified in the context, and participate in decision making. A decision making model overstepping the limits of the three-phase Simon's model is offered. The paper proposes a set of technologies that can be used to implement the ideas behind the research. An application of these ideas is illustrated by an example of usage of the developed DSS for planning fire response actions.

Keywords Context-aware decision support · Context model · Decision making model · Service-oriented architecture · Emergency response · Ridesharing

A. Smirnov (✉) · T. Levashova · N. Shilov · A. Kashevnik
St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, St. Petersburg, Russia
e-mail: smir@iias.spb.su

T. Levashova
e-mail: Tatiana.levashova@iias.spb.su

N. Shilov
e-mail: nick@iias.spb.su

A. Kashevnik
e-mail: alexey@iias.spb.su

A. Smirnov
ITMO University, St. Petersburg, Russia

M.S. Obaidat et al. (eds.), *Simulation and Modeling Methodologies, Technologies and Applications*, Advances in Intelligent Systems and Computing 319,
DOI 10.1007/978-3-319-11457-6_1

1 Introduction

Operational decision support is required in situations happened in dynamic, rapidly changing, and often unpredictable distributed environments. Such situations can be characterized by highly decentralized, up-to-date data sets coming from various resources. The goals of context-aware support to operational decision making are to timely provide the decision makers with up-to-date information, to assess the relevance of information and knowledge to the decision, and to gain insight in seeking and evaluating possible decision alternatives.

The present research addresses theoretical and technological foundations of context-aware operational decision support in dynamic environments. The theoretical foundations are built around ontologies. The ontologies are a widely accepted means for context information modeling. They provide efficient facilities to represent application knowledge and to enable the resources of the dynamic environments to be context-aware and interoperable. The proposed fundamentals are supported by advanced intelligent technologies with their application to the Web.

The decision making model used in the research follows the three-phase model proposed by Simon [14]. The used model oversteps the limits of the three-phase model towards automatic search for an efficient workable decision and communications on the decision implementation.

The ideas behind the research are incorporated into a context-aware decision support system (DSS). The developed DSS has service-oriented architecture. Such architecture facilitates the interactions of service components and the integration of new ones [1, 12, 19]. The system is intended to support decisions on involvement of independent parties in the joint actions according to the current situation and scheduling these actions.

The rest of the paper is structured as follows. Section 2 proposes theoretical foundations to be followed when building context-aware DSSs. Section 3 presents technologies supporting the proposed theoretical foundations and service-oriented architecture of the developed DSS. Application of the presented ideas to the emergency management domain is illustrated in Sect. 4. Main research results are summarized in the conclusion.

2 Theoretical Foundations

Decision support in the dynamic environments has to take into account constant environmental changes. In the present research, resources of the environment provide information of any changes to the DSS. These resources are referred to as *information resources*. The information resources perform the needed computations and solve problems, as well. The collection of information resources comprises various kinds of sensors, electronic devices, databases, services, etc. Besides information resources, the research distinguishes one more type of resources that is *acting resources*. These resources include people and/or organizations that can be involved in the joint actions.

The research follows the knowledge-based methodology to building DSSs. The idea behind the research is to represent the application knowledge by means of constraints. This knowledge is described using two independent sorts of reusable components: domain ontology and task ontology. The domain ontology represents conceptual knowledge about the application domain. The task ontology describes problems occurring in the application domain and methods for achieving solutions to these problems (problem-solving methods). The both components make up the application ontology, which is represented as a set of constraints. This ontology specifies non-instantiated knowledge.

The resources' representations are supposed to be compatible with the ontology representation. The application ontology and the resources' representations are aligned. The alignment indicates what information resource(s) instantiates the given property of the given object specified in the ontology.

In the research, context model serves to represent the knowledge about a decision situation (the settings in which decisions occur and the problems requiring solutions). Context is suggested being modeled at two levels: abstract and operational. These levels are represented by abstract and operational contexts, respectively (Fig. 1).

Abstract context is an ontology-based model integrating information and knowledge relevant to the current decision situation. The DSS's user (the decision maker) in his/her request to the DSS indicates the type of the current situation or smart sensors provide this type to the system. The relevant information and knowledge are extracted from the application ontology. As the two components make up this ontology, the abstract context specifies domain knowledge describing the current situation and problems to be solved in this situation.

The abstract context reduces the amount of knowledge represented in the application ontology to the knowledge relevant to the decision situation. In the application ontology this knowledge is related to the resources via the alignment, therefore the abstract context allows the set of resources to be reduced to the

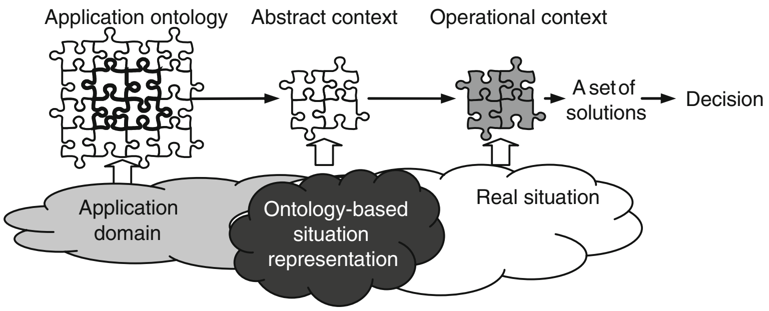


Fig. 1 Context-aware decision support

resources needed to instantiate knowledge specified in the abstract context. The reduced set of resources is referred to as contextual resources.

Operational context is an instantiation of the domain constituent of the abstract context with data provided by the contextual resources. This context reflects any changes in environmental information, in this way it is a near real-time picture of the current situation. The operational context embeds the constraint-based specifications of the problems to be solved. Those input parameters of the problems, which correspond to properties of the objects specified in the domain constituent, are instantiated.

The embedded in the operational context problems are processed as a constraint satisfaction problem in its enumeration form. As a result, a set of feasible alternative 'satisfactory' solutions in the current situation is produced. Each solution is a plan of joint actions for the acting resources in the current situation. Decision making is regarded as a choice between the alternatives.

If one or more efficiency criteria are applied to the set of feasible solutions an efficient solution can be found. The efficient solution is considered as the workable decision. The acting resources included in the efficient plan communicate with the DSS in the person of the decision maker on acceptance/rejection of this plan, i.e. on the plan implementation.

In order to enable capturing, monitoring, and analysis of the implemented decisions and their effects the abstract and operational contexts with references to the respective decisions are retained in an archive. As a result, the DSS is provided with reusable models of decision situations. These models, for instance, are used to reveal user preferences based on the analysis of the operational contexts in conjunction with the implemented decisions.

Search for a 'satisfactory' decision is the main principle of the decision making model proposed by Simon [14]. The main conclusions from the Simon's investigation of decision making process can be formulated as follows: the efforts of decision makers to evaluate consequences of the possible alternatives and dependence of the decision makers on the multiple factors influencing their choice should be minimized. Simon proposed a 'satisfactory' decision as a result of decision making. That is a decision that is neither efficient nor optimal, but the decision that satisfies all the stakeholders interested in it.

The presented here constrained-based approach enables to express the multiple influencing factors (e.g., the preferences of the stakeholders interested in the decision, intervals of the resources' availabilities, the resources' costs, etc.) by means of constraints. The constraints formalizing the multiple factors along with the constraints specified in the operational context are processed as a constraint satisfaction problem. A set of feasible (satisfactory) plans is the result of problem solving. At that, these plans do not depend on decision makers' attentions, information they have, or stress. Moreover, the decision makers are saved from information overload. The decision maker can choose any plan from the set or take advantage of some predefined efficiency criteria.

The Simon's model specifies decision making consisting of "intelligence", "design", and "choice" phases. Table 1 shows the correspondences between the

Table 1 Three-phase model

Phase	Phase content	Steps of Simon's model	Steps of constraint-based approach
Intelligence	Finding, identifying, and formulating the problem or situation that calls for a decision	Fixing goals	Abstract context creation
		Setting goals	Operational context producing
Design	Search for possible decisions	Designing possible alternatives	Constraint-based generation of feasible alternative satisfactory solutions
Choice	Evaluation of alternatives and choosing one of them	Choice of a satisfactory decision	Choice of an efficient satisfactory decision
Implementation	Putting the decision into action	–	Communications of the actors with the DSS for their actions

steps of the constrained-based approach and the phases of the Simon's model. The proposed approach exceeds the bounds of the Simon's model proposing two more steps: search for an efficient satisfactory decision and communications on the implementation of this decision.

In the following Section, technologies supporting the proposed theoretical foundations are presented.

3 Technological Foundations

According to the theoretical foundations, the application ontology should be represented by means of constraints. Knowledge specified in such a way can be treated as a constraint satisfaction problem. The formalism of object-oriented constraint networks (OOCN) [15] is proposed to be used for the ontology representation. This formalism supports object-oriented knowledge representation (class-attribute-range modeling) and allows specification of the following sets of constraints: (1) *taxonomical* ("is-a") relationships, (2) *hierarchical* ("part-of") relationships, (3) *class cardinality* restrictions, (4) *class compatibility* relationships, (5) *associative* relationships, and (6) *functional* relations.

DSSs intended for the dynamic environments become more efficient when they take advantage of Web service technologies. Hence, it is proposed to implement the DSS as a service-oriented system. Web services can support and simplify the exchange of context information; they enable service-oriented systems to utilize various types of context information to adapt their behaviors and operations to dynamic changes. Service-oriented architecture can support integration and consolidation of activities of independent actors.

In the research, capabilities of various resources (resource functionalities) and their delivery constraints are made available through the Web-services [21]. These capabilities and constraints are captured by a service profile [8]. A profile describes the functional and non-functional service semantics. The functional service semantics is described in terms of the input and output parameters of the service. The non-functional service semantics is described with respect to service's cost model, availability, competence, and weight.

In the ontology-based service-oriented systems the idea of the alignment of the resource representations against an ontology is implemented using methods of semantic matching [8]. In the present research the Web-services' descriptions in WSDL [20] are aligned against the application ontology. As the result, the WSDL-descriptions are complemented with appropriate SA-WSDL [13] annotations. The alignment is based on semantic matching between the application ontology and the WSDL-descriptions. An approach combining three classes of matching methods—combined, linguistic, and contextual—is used for the matching [16]. The alignment enables Web-services to be interoperable and facilitates Web-service composition.

The problem of relevant knowledge determination in the ontology-based systems is treated as slicing operation. The purpose of this operation is to extract pieces of knowledge from one or more ontologies, which considered as relevant to the user request. The implementation of the operation depends on the ontology representation formalism (see e.g., [3, 18]).

In the present research, the slicing operation is based on the determination in the application ontology knowledge semantically similar to the user request. The operation captures knowledge related to the semantically similar knowledge based on attribute inheritance and constraints processing rules [11]. The captured knowledge is extracted and integrated into the abstract context. In terms of OOCN the abstract context is a set of constraints formalizing (1) the domain knowledge to be instantiated to produce the picture of the current situation and (2) the situation's problems in general form. According to the alignment associations, the set of Web-services is reduced to the contextual ones.

In the part of the abstract context instantiation by the resources, the research follows the idea to integrate these resources based on the service composition [2, 7, 8]. Web-service composition is the act of taking several semantically annotated component services, and bundling them together to meet the needs of a given customer [9].

In the present research the customer needs are formalized in the abstract context. This context specifies an abstract workflow of the required composite Web-service. Contextual Web-services participate in the composition. They communicate in terms of their inputs/outputs to create a service execution sequence. If alternative services available a set of sequences is created. A specific alternative is chosen based on the principles of maximum functionality, maximum access interval, and minimum service weight [17].

The contextual Web-services produce an operational context. The used knowledge representation by means of the OOCN-formalism is compatible with representations supported by constraint solvers. Therefore, the operational context can be processed as a constraint satisfaction problem using such solvers. Some of these solvers provide mechanisms to search for optimal or efficient solutions, e.g., ILOG [5].

The acting resources communicate on the decision implementation using wire- or wireless Internet-accessible devices.

Table 2 summarizes the technologies supporting the proposed here concept.

The service-oriented architecture of the DSS comprises three groups of services (Fig. 2). The first group is made up of core services responsible for the registration of the Web-services in the service register and producing the real-world model of

Table 2 Technological framework

Objectives in the theoretical foundations	Techniques	Technology	Result in terms of OOCN
Application ontology building	Ontology building from scratch, integration of existing ontologies	Ontology engineering, ontology management	OOCN with non-instantiated variables
Resource representation	Service-based representation	Semantic Web-services	OOCN with non-instantiated variables
Overcoming Web-services heterogeneity	Alignment of ontology and service descriptions	Semantic matching	OOCN with associative (alignment) constraints
Abstract context creation	Ontology slicing	Ontology management	OOCN with non-instantiated variables
Determination of contextual resources	Ontology slicing	Ontology management	OOCN with associative (alignment) constraints
Operational context producing	Service communications	Web-service composition, context management, constraint satisfaction	OOCN with partly instantiated variables
Generation of alternative action plans	Solving of constraint satisfaction problem	Constraint satisfaction	OOCN with fully instantiated variables, a set of feasible solutions
Choice of a specific plan	Optimization	Constraint programming	An efficient solution
Plan implementation	Service communications	Mobile applications, collective decision making	The efficient solution
Context reusability	Context archiving	Context management	OOCN with partly instantiated variables
Revealing user preferences	Context-based decision archiving	Profiling, decision mining	A set of user constraints
DSS implementation	Service-oriented architecture	Web-services	–

the decision situation, i.e. the creation of the abstract and operational contexts. Services belonging to this group are as follows:

- *registration service* registers the Web-services in the service register;
- *application ontology service* provides access to the application ontology;
- *abstract context service* creates, stores, maintains, and reuses the abstract contexts;
- *operational context service* produces the operational contexts.

Web-services comprising the second group are responsible for the generation of alter-native plans for actions and the selection of an efficient plan. This group contains:

- *composition service* coordinates the service composition process;
- *constraint satisfaction service* generates sets of feasible plans for actions;
- *decision making* service selects efficient plan(s) from the sets of feasible plans, coordinates the communications between the acting resources and the plan implementation.

The third group comprises sets of services responsible for the representation of the resources and implementation of their functions. This group includes:

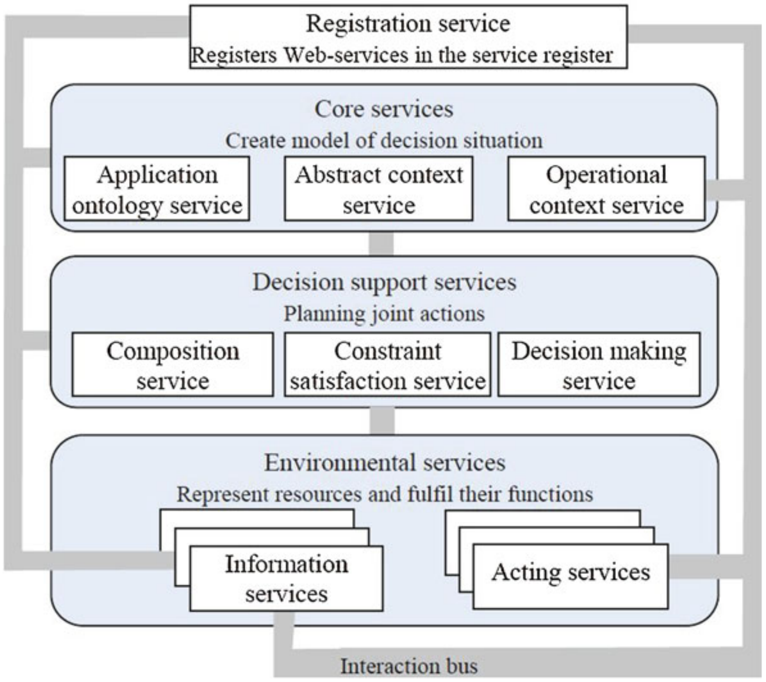

Registration service
Registers Web-services in the service register
Core services
Create model of decision situation
Application ontology service
Abstract context service
Operational context service
Decision support services
Planning joint actions
Composition service
Constraint satisfaction service
Decision making service
Environmental services
Represent resources and fulfil their functions
Information services
Acting services
Interaction bus

Fig. 2 Service-oriented architecture of DSS

- *information services* provide data stored in the resources' profiles and implement functions of the information resources;
- *acting services* provide data stored in the profiles of the acting resources; represent roles played by people or organizations; communicate on the plan implementation.

4 Fire Response

The DSS to support decisions on planning joint actions of independent parties has been developed based on the presented here foundations. The system is used to support decisions in emergency scenarios. Emergencies are a good example of ever-changing situations.

This Section illustrates the system usage in the simulated fire scenario. The purposes of the DSS in the fire situation are to produce a fire response plan for emergency responders, to offer an evacuation plan for potential victims, and to support the communications on decision implementation between the parties involved in the plans. Independent parties participating in the implementation of the fire response plans organize the fire response community (Fig. 3).

Figure 4 presents the abstract context created to model the fire situation. This context is created from the application ontology of the emergency management

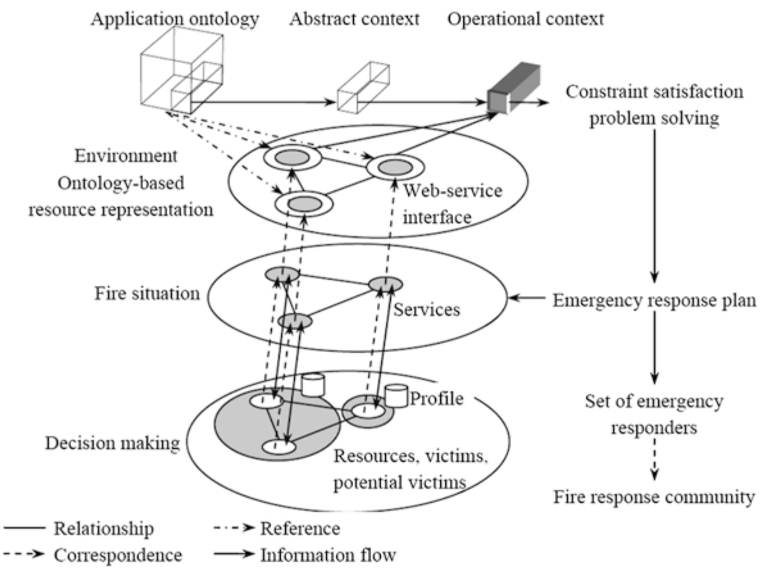


Fig. 3 Organization of fire response community

domain. The ontology has 7 taxonomy levels, contains more than 600 classes, 160 class attributes, and 120 constraints of different types. The abstract context has 4 taxonomy levels, contains 19 bottom-level classes to be instantiated, 38 class attributes, and around 30 constraints.

The abstract context represents the following kinds of acting resources needed in the response actions: fire brigades, emergency teams, hospitals, and car drivers. As well, this context represents kinds of transportation devices the mobile acting resources can go by. The context specifies (not shown by the taxonomy) that the emergency teams can go by ambulances and rescue helicopters, the fire brigades can go by fire engines and fire helicopters, and the car drivers go by cars.

Task knowledge is hidden in the class "emergency response". This class specifies the following problems:

- select feasible hospitals, emergency teams, fire brigades, and car drivers;
- determine feasible transportation routes for ambulances and fire engines depending on the transportation network and traffic situation;
- calculate the shortest routes for transportation of the emergency teams by ambulances, fire brigades by fire engines, and evacuees by cars;

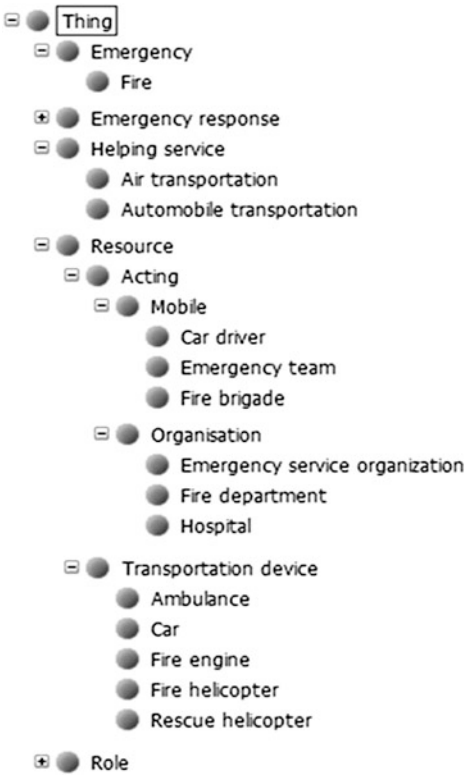


Fig. 4 Fire situation: abstract context (a piece)

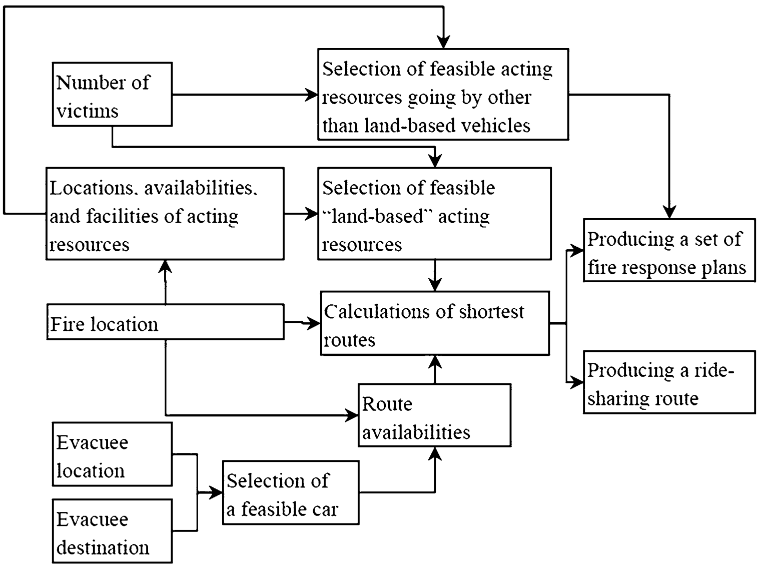

Number of victims
Selection of feasible acting resources going by other than land-based vehicles
Locations, availabilities, and facilities of acting resources
Selection of feasible "land-based" acting resources
Producing a set of fire response plans
Fire location
Calculations of shortest routes
Producing a ride-sharing route
Route availabilities
Evacuee location
Evacuee destination
Selection of a feasible car

Fig. 5 Execution sequence of information web-services

- produce a set of feasible response plans for emergency teams, fire brigades, and hospitals;
- produce a set of feasible ridesharing routes.

The execution sequence of Web-services composed to instantiate the abstract context and solve the specified problems is shown in Fig. 5.

In the simulated scenario it is supposed that 9 injured people have to be transported to hospitals. In the territory adjacent to the fire place 7 available fire brigades, 8 emergency teams, 5 hospitals having free capacities for 4, 4, 2, 3, and 3 patients are found; 6 fire trucks and 1 fire helicopter are allocated to the fire brigades, 7 ambulances and 1 rescue helicopter are allocated to the emergency teams; 1 fire brigade is calculated to be required to extinguish the fire. The plan for actions designed for the emergency teams supposes that one vehicle can house one injured person.

The set of feasible plans for actions is generated for the criteria of minimal time and cost of transportation of all the victims to hospitals, and minimal number of mobile emergency responders involved in the response actions. The set of plans comprises 4 plans. The efficient plan is selected based on the key indicator of minimal time of victim transportations.

Figure 6 presents the operational context and the efficient plan for actions for the emergency teams, fire brigades, and hospitals, i.e. for professional emergency responders. Such a plan is a set of emergency responders with transportation routes for the mobile responders, required helping services, and schedules for the responders' activities. In Fig. 6 the big dot denotes the fire location; the dotted lines

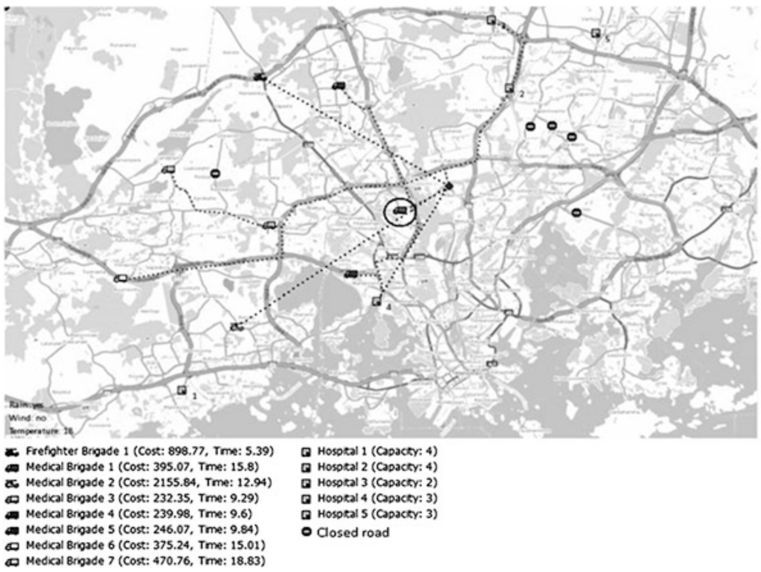

Rain: yes
Wind: no
Temperature: 18
Firefighter Brigade 1 (Cost: 898.77, Time: 5.39)
Medical Brigade 1 (Cost: 395.07, Time: 15.8)
Medical Brigade 2 (Cost: 2155.84, Time: 12.94)
Medical Brigade 3 (Cost: 232.35, Time: 9.29)
Medical Brigade 4 (Cost: 239.98, Time: 9.6)
Medical Brigade 5 (Cost: 246.07, Time: 9.84)
Medical Brigade 6 (Cost: 375.24, Time: 15.01)
Medical Brigade 7 (Cost: 470.76, Time: 18.83)
Hospital 1 (Capacity: 4)
Hospital 2 (Capacity: 4)
Hospital 3 (Capacity: 2)
Hospital 4 (Capacity: 3)
Hospital 5 (Capacity: 3)
Closed road

Fig. 6 Plan for actions for professional emergency responders

designate the routes proposed for the transportations of the emergency teams and fire brigades. One ambulance (encircled in the figure) and the rescue helicopter go from the fire location to hospitals twice. The estimated time of the operation of transportations of all the victims to hospitals is 1 h. 25 min.

The interaction sequence of the architectural Web-services for producing the efficient plan for the professional emergency responders is shown in Fig. 7.

The efficient plan is presented to the emergency responders for their decision making on the plan implementation. They can access the operational context using any Internet-accessible devices (notebooks, PDAs, mobile phones, etc.). Figure 8 shows part of the plan displayed on the Tablet PC of the leader of an emergency team going by ambulance. He/she can accept or reject this plan (a special option is provided for this). The option of rejection is provided for due to the rapidly changing emergency situations—something may happen between the moment when the plan is selected and the time when the emergency responders receive this plan.

The procedure of making decisions by the professional emergency responders is as follows (Fig. 9). If the plan is approved by all the responders, this plan is supposed to be the plan for actions. Otherwise, either this plan is adjusted (so that the potential participant who refused to act according to the plan does not appear in the adjusted plan) or another set of plans is produced.

The plan adjustment is a redistribution of the actions among emergency responders that are contained in the set of feasible plans. If such a distribution does not lead to a considerable loss of time (particularly, the estimated time of the

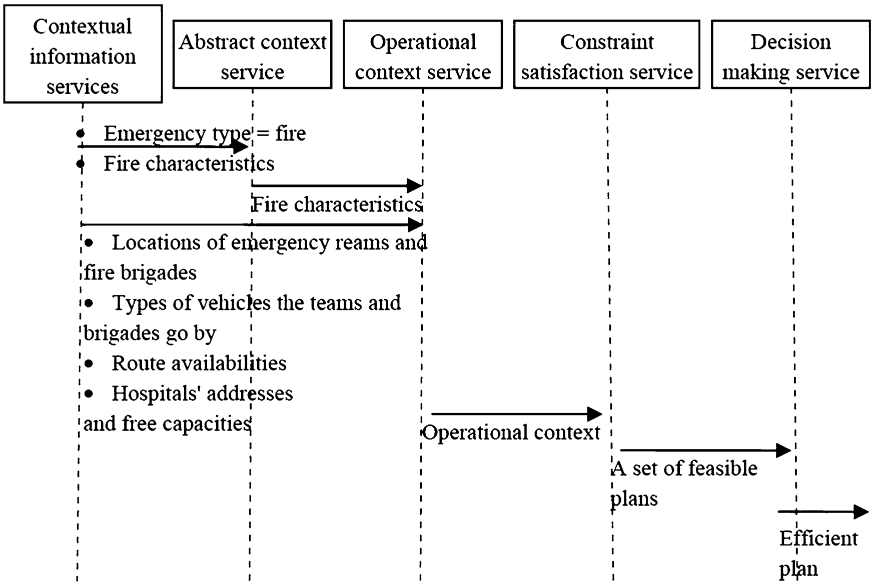

Contextual information services
Abstract context service
Operational context service
Constraint satisfaction service
Decision making service
• Emergency type = fire
• Fire characteristics
Fire characteristics
• Locations of emergency reams and fire brigades
• Types of vehicles the teams and brigades go by
• Route availabilities
• Hospitals' addresses and free capacities
Operational context
A set of feasible plans
Efficient plan

Fig. 7 Service interactions for planning actions of professional emergency responders

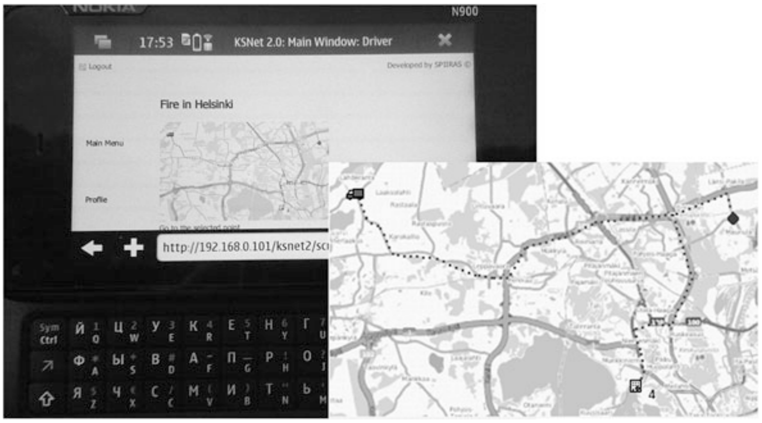


Fig. 8 Plan for actions for an emergency team

transportation of the injured people to hospitals does not exceed "The Golden Hour" [10]) then the adjusted plan is submitted to the renewed set of emergency responders for approval. If a distribution is not possible or leads to loss of response time a new set of plans is produced, from which a new efficient plan is selected and submitted to approval. Simultaneously with planning the joint actions for professional emergency responders, the evacuation activities are planned. Features provided by the ridesharing technology are used to organize the evacuation of potential

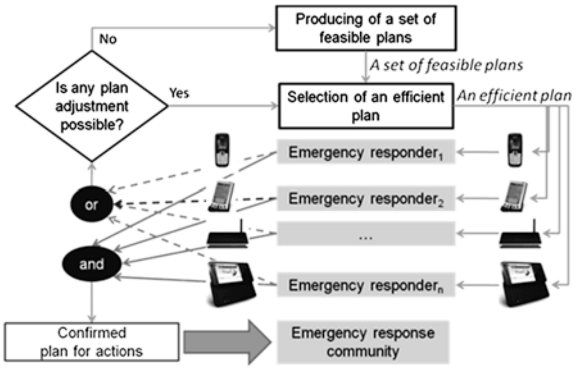

Producing of a set of feasible plans
A set of feasible plans
No
Is any plan adjustment possible?
Yes
Selection of an efficient plan
An efficient plan
Emergency responder$_1$
Emergency responder$_2$
...
Emergency responder$_n$
or
and
Confirmed plan for actions
Emergency response community

Fig. 9 Decision making by professional emergency responders

victims. Potential victims here are people who have been out of danger so far or have got themselves out of the dangerous area. Success of the evacuation operation heavily depends on the availability of volunteers as drivers of the passing cars.

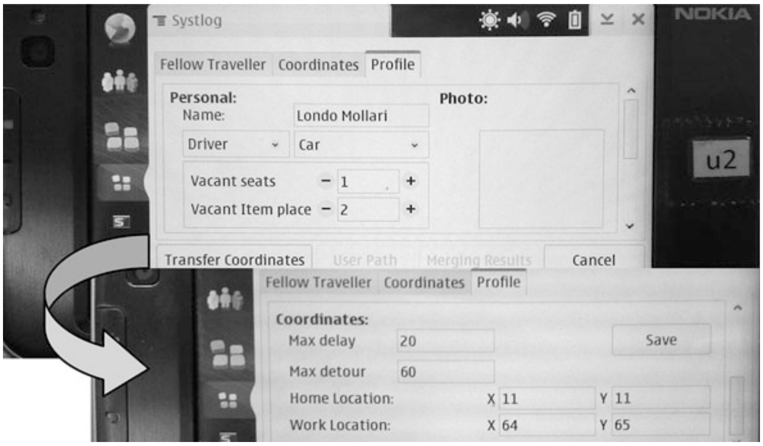
Persons who need to be evacuated invoke a ridesharing service that is responsible for the evacuation. Clients of this service are supposed to be installed on the Internet-accessible devices of car drivers and other people involved in the fire situation. The persons enter the locations they would like to be conveyed. The ridesharing service determines the persons' locations and searches for cars going to or by the same or close destinations that the persons would like to be. It searches the cars among the vehicles passing the persons' locations. This service reads information about the destinations that the car drivers are going to from the navigators that the drivers use or from the drivers' profiles. The profiles store periodic routes of the drivers and cars' properties as the number of passenger seats, the availabilities of baby car seats, etc. (Fig. 10).

Based on the information about the locations and destinations of the person and the found cars a set of feasible routes for the person transportations is generated. An efficient route is determined based on the criterion of minimum transportation time.

The ridesharing service sends appropriate signals to the drivers included in the ridesharing routes and displays on the drivers' devices the routes each driver is selected for. The points where the driver is expected to pick up the passenger(s) is indicated in the routes. The ways the passengers have to walk to these points are routed for them as well. Besides the routes, the passengers are informed of the model, color, and license plate number of the car intended for their transportation. The persons that cannot be evacuated by passing cars are informed that they can be evacuated by taxi. If they agree, the ridesharing service makes orders for taxi.

Generally speaking, the destinations for the evacuated people do not matter. In actual usage the evacuees can just run the ridesharing service and it will search for passing cars.

Decision making on an evacuation plan is in making agreement between the driver and the evacuee to go according to the scheduled ridesharing route (Fig. 11). In case, when there is no agreement between a driver and an evacuee, another car



Personal:
Name: Londo Mollari
Driver Car
Photo:
Vacant seats 1
Vacant Item place 2

Coordinates:
Max delay 20
Max detour 60
Home Location: X 11 Y 11
Work Location: X 64 Y 65

Fig. 10 Driver's profile

for evacuation of this passenger is sought for. At that, the confirmed routes are not revised.

In the considered scenario results of evacuation using the ridesharing technology are as follows: 26 persons desire to be evacuated from the scene of fire; 22 persons have been driven directly to the destinations by 16 cars whereas for 4 persons no cars have been found. Examples of ways routed for a driver and a passenger are given in Figs. 12 and 13. The encircled car in the figures shows the location where the driver is offered to pick up the passenger.

The interaction sequence of Web-services for producing a ridesharing route is shown in Fig. 14. The boxes "Evacuee" and "Car driver" mean Web-services representing the evacuees and car drivers, respectively.

The Smart-M3 platform [4] was used in the execution of the fire response scenario. Tablet PC Nokia N810 (Maemo4 OS), smart phone N900 (Maemo5 OS),

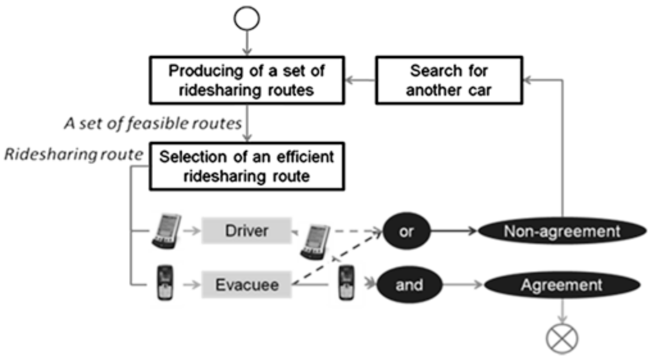


Fig. 11 Decision making by car drivers and evacuees

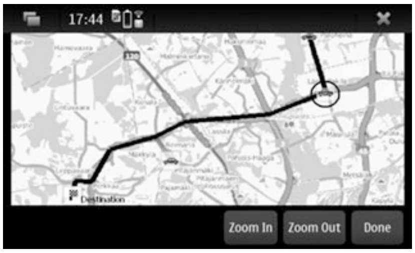


Fig. 12 Ridesharing route: driver's view

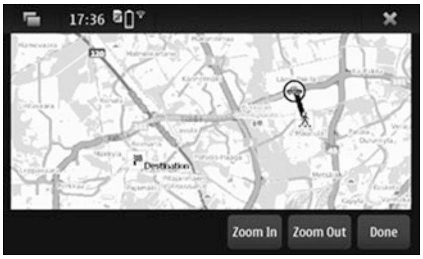


Fig. 13 Ridesharing route: passenger's view

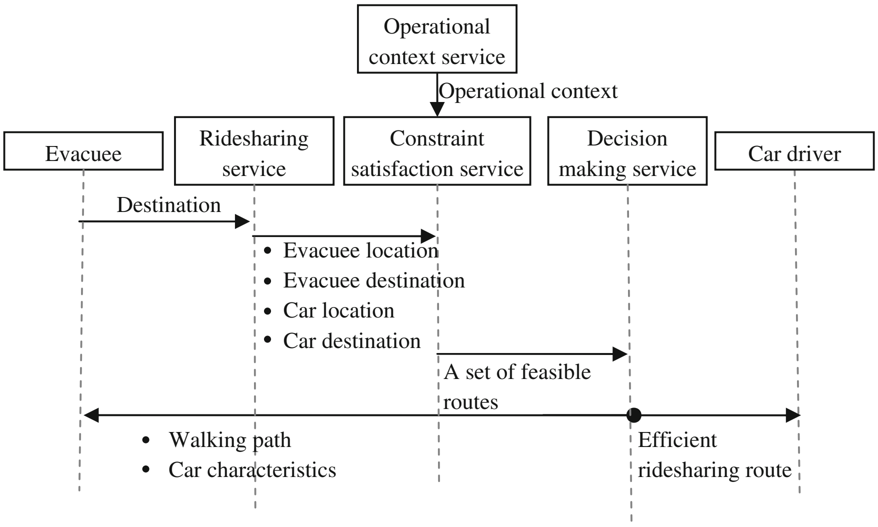


Fig. 14 Service interactions for evacuation planning

and different mobile phones served as the user devices. Personal PCs based on Pentium IV processors and running under Ubuntu 10.04 and Windows XP were used for hosting other services.

5 Conclusions

Theoretical and technological foundations for building context-aware DSSs intended for the dynamic environments are proposed. The theoretical foundations offer a concept of context-aware DSS. According to this concept the DSS uses ontology-based context model formalized as a set of constraints. The technological foundations propose a set of technologies enabling Web-based implementation of the theoretical ideas.

The constraint-based context representation allows the DSS to process the problems specified in the context along with other constraints possibly influencing the decision maker's choice as a constraint satisfaction problem. As the result, the DSS provides the decision maker with a workable satisfactory decision. Such a decision is the main principle of the Simon's decision making model. The presented approach exceeds the bounds of this model towards the automatic search for an efficient workable decision and the actors' communications on the decision implementation. It can be concluded that the constraint-based approach enables to minimize the efforts of decision makers to evaluate consequences of the possible alternatives and dependence of the decision makers on the multiple factors influencing their choice.

The main ideas behind the research are illustrated via the support of decisions on planning fire response actions. The response actions are considered comprising two simultaneous scenarios: (1) the response from the professional emergency responders, which is purposed on transportation of the injured people to hospitals and extinguishing the fire; and (2) the response from the volunteers agreeing to evacuate the potential victims. The problem of planning actions of the professional emergency responders is treated as a dynamic logistic problem. The problem of planning volunteers' actions is processed as a ridesharing problem. The constraint satisfaction technology is used for problem solving. This technology enables decision support systems to automatically provide decision makers with efficient 'satisfactory' solutions.

Acknowledgments The present research was partly supported by the projects funded through grants 12–07–00298, 13–07–00336, 13–07–12095, 13–07–13159, 14–07–00345, 14–07–00427 (the Russian Foundation for Basic Research), the Project 213 (the research program "Information, control, and intelligent technologies & systems" of the Russian Academy of Sciences (RAS)), the Project 2.2 (the Nano- & Information Technologies Branch of RAS), and grant 074–U01 (the Government of the Russian Federation).

References

1. Alonso G, Casati F, Kuno HA, Machiraju V (2004) Web services—concepts, architectures and applications. Springer, Heidelberg Berlin
2. Bangemann T, Diedrich C, Riedl M et al (2009) Integration of automation devices in web service supporting systems, preprints of the 30th IFAC workshop on real-time programming and 4th international workshop on real-time software, pp 161–166

3. Chaudhri VK, Lowrance JD, Stickel ME et al (2000) Ontology construction toolkit. Technical note ontology, AI Center. Report, 2000. SRI Project no. 1633
4. Honkola J, Laine H, Brown R, Tyrkko O (2010) Smart-M3 information sharing platform. In: Proceedings of IEEE symposium computers and communications, IEEE computer society, pp 1041–1046. doi:http://ieeecomputersociety.org/10.1109/ISCC.2010.5546642
5. IBM ILOG, Web (2011) URL: http://www-01.ibm.com/software/websphere/ilog/. Accessed 27 Feb 2014
6. Jammes F, Mensch A, Smit H (2005) Service-oriented device communications using the devices profile for web services, proceedings of the 3rd international workshop on middleware for pervasive and ad-hoc computing (MPAC'05), ACM. http://doi.acm.org/10.1145/1101480.1101496
7. Karnouskos S (2011) Realising next-generation web service-driven industrial systems. Int J Adv Manuf Technol. doi:10.1007/s00170-011-3612-z (Springer)
8. Klusch M (2008) Semantic web service description. In: Helin H, Schuldt H (eds) Intelligent service coordination in the semantic web. Birkhaeuser Verlag, Springer, pp 41–67
9. Klusch M (2008) Semantic web service coordination. In: Helin H, Schuldt H (eds) Intelligent service coordination in the semantic web. Birkhaeuser Verlag, Springer, pp 69–108
10. Lerner EB, Moscati RM (2001) The golden hour: scientific fact or medical "urban legend?". Acad Emerg Med 8(7):758–760
11. Levashova T (2008) Context model in intelligent decision support systems, decision support. In: Petrovsky A (ed) Proceeding of institute of systems analysis of the Russian academy of sciences, Moscow, URSS 35, pp 33–42 (in Russian)
12. Papazoglou MP, van den Heuvel W-J (2007) Service oriented architectures: approaches, technologies and research issues. VLDB J 16(3):389–415
13. Semantic Annotations for WSDL and XML Schema, W3C Recommendation, Web (2007) URL: http://www.w3.org/TR/sawsdl/. Accessed 27 Feb 2014
14. Simon HA (1987) Making management decisions: the role of intuition and emotion. Acad Manage Exec 1:57–64
15. Smirnov A, Pashkin M, Chilov N, Levashova T, Haritatos F (2003) Knowledge source network configuration approach to knowledge logistics. Int J Gen Syst 32(3):251–269
16. Smirnov A, Kashevnik A, Shilov N, Balandin S, Oliver I, Boldyrev S (2010) On-the-Fly ontology matching in smart spaces: a multi-model approach, smart spaces and next generation wired/wireless networking. In: Balandin S, Dunaytsev R, Koucheryavy Y (eds) Proceedings of the 3rd conference ruSMART 2010 and the 10th international conference NEW2AN 2010, St. Petersburg, Russia, 2010, Springer, LNCS 6294, pp 72–83
17. Smirnov A, Levashova T, Shilov N, Kashevnik A (2010) Hybrid technology for self-organization of resources of pervasive environment for operational decision support. Int J Artif Intell Tools 19(2):211–229. doi:http://dx.doi.org/10.1142/S0218213010000121 (World Scientific Publishing Company)
18. Swartout B, Patil R, Knight K, Russ T (1996) Toward distributed use of large-scale ontologies, tenth knowledge acquisition for knowledge-based systems workshop (KAW'96), Banff, Canada, 1996, Web. URL: http://ksi.cpsc.ucalgary.ca/KAW/KAW96/swartout/Banff_96_final_2.html. Accessed 27 Feb 2014
19. Web Services Architecture, W3C Working Group Note, Web (2004) URL: http://www.w3.org/TR/ws-arch/. Accessed 27 Feb 2014
20. Web Services Description Language (WSDL), W3C Note, Web (2001) URL: http://www.w3.org/TR/wsdl. Accessed 27 Feb 2014
21. Web Services on Devices, Microsoft Corp. (2007) Web, URL: http://msdn.microsoft.com/-en-us/library/bb756908.aspx. Accessed 27 Feb 2014